JANUARY/FEBRUARY 1977 $1.50

MICROTREK

THE MICROCOMPUTER MAGAZINE FOR THE HOBBYIST & PROFESSIONAL

How to interface a microcomputer to a pocket calculator

Buyer's Review: the Digital Group System

Modular program design

Assembly Language Translation – Part II

SWTPC 6800-SOFTWARE

WARNING — It has been determined that reading this ad may be hazardous to your health, if you own another type computer system. We will not be responsible for ulcers, heartburn, or other complications if you persist in reading this material.

4 K BASIC© — 8 K BASIC©

* Full floating point math
* 1.0E-99 to 9.99999999E+99 number range
* User programs may be saved and loaded
* Direct mode provided for most statements
* Will run most programs in 8K bytes of memory (4K Version)
  or 12K bytes of memory (8K Version)
* USER function provided to call machine language programs
* String variables and trig functions—8K BASIC only

COMMANDS

LIST
RUN
NEW
SAVE
LOAD
PATCH
REM
DIM
DATA
READ
RESTORE
LET*
FOR

STATEMENTS

END
GOTO*
ON. . .GOTO*
ON. . .GOSUB*
IF. . .THEN*
INPUT
PRINT*
NEXT
STOP
GOSUB*
PATCH*
RETURN
† DES
† PEEK
† POKE

FUNCTIONS

ABS
INT
RND
SGN
CHR
USER
TAB
† VAL
† EXT$
† LEN$
† LEFT$
† MID$
† RIGHT$
† SIN
† COS
† TAN
† EXP
† LOG
† SQR

** Direct mode statements*
† 8K Version only

MATH OPERATORS

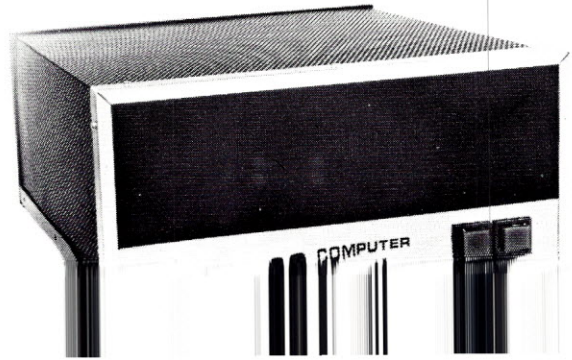
- (unary) Negate
* Multiplication
/ Division
+ Addition
– Subtraction
† ↑ Exponent

RELATIONAL OPERATORS

= Equal
<> Not Equal
< Less Than
> Greater Than
<= Less Than or Equal
>= Greater Than or Equal

Your BANKAMERICARD welcome here

WE HONOR master charge THE INTERBANK CARD



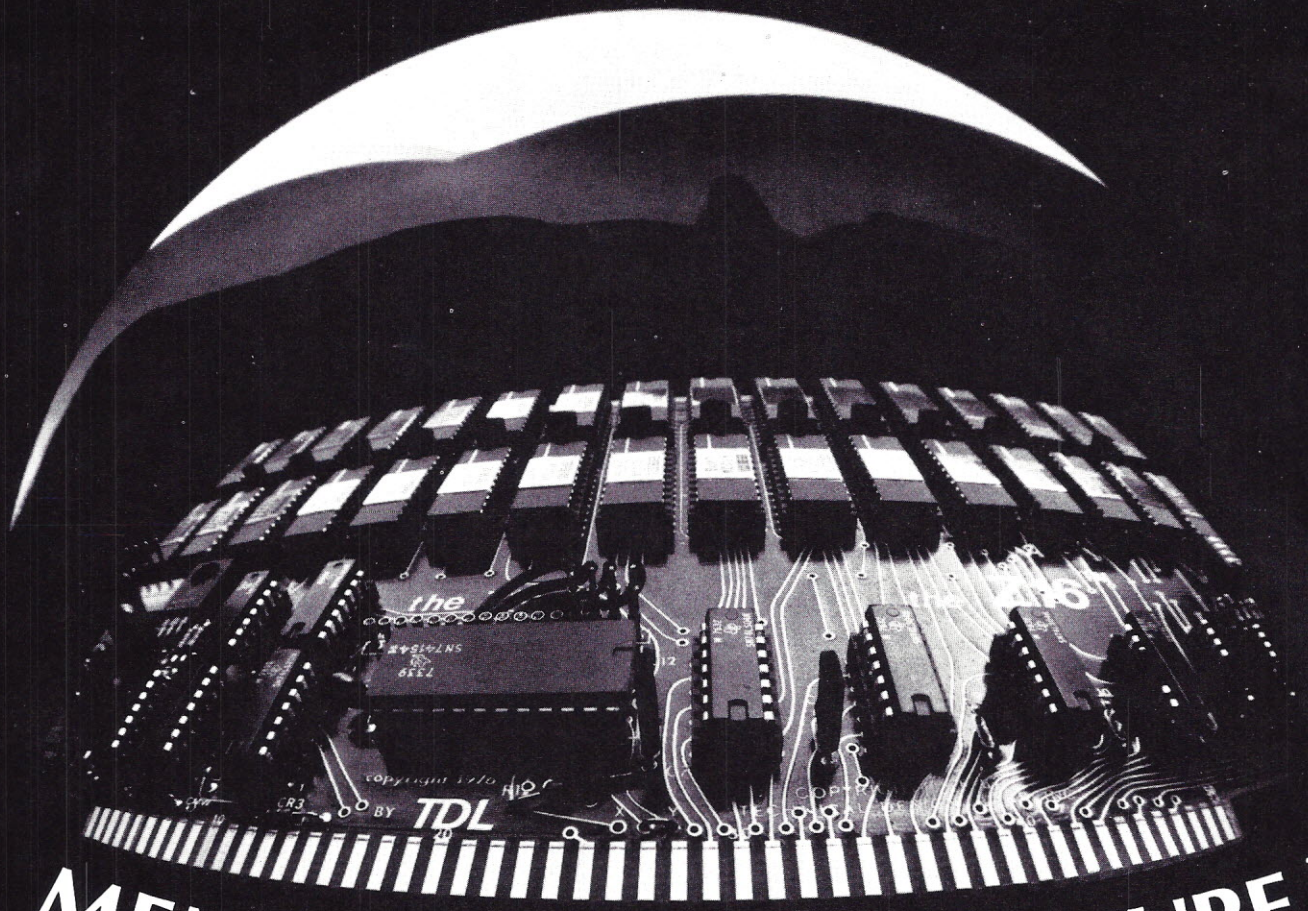
You guys are out of your minds, but who am I to complain. Send —

☐ 4K BASIC CASSETTE $4.95 ☐ MP-68 Computer
☐ 8K BASIC CASSETTE $9.95 Kit $395.00

NAME

ADDRESS

CITY STATE ZIP



MEMORY FROM THE FUTURE*

Z16

COMBINING:
- HIGHEST DENSITY
- FASTEST ACCESS
- LOWEST POWER USE
- HIGHEST QUALITY

- GREATEST VERSATILITY
- S100 BUS COMPATIBLE
- LOWEST COST 16K STATIC MEMORY MODULE AVAILABLE

Full 16K of memory on one card available in 4K increments. Buy only what you need now. Expansion later is easy with a board you have already tested.

Utilizes the EMM SEMI 4200 memory chip which is organized as 4K by 1 bits. Provides maximum access time of only 200ns. Added to board logic time, total board access time is below 250ns. No other memory board made to S100 bus specs can match this.

Each 4K block may be individually addressed at any 4K page border. You have the versatility of most 4K boards in your 16K package. Address changes are very easily accomplished by using a simple jumper scheme. Each 4K block may be individually protected by a switch.

Power consumption is outstandingly low! Only 205ma from the +8v, 105ma from the +16v, and 24ma from the −16v, for a FULL 16K. Battery backup with a simple jack connector.

Fully solder masked and silk screened board, sockets for all IC's. Complete documentation includes source code for comprehensive memory test program and paper tape of this program.

KIT: 4K - $169; 8K - $295; 12K - $435; 16K - $574; 4K expansion kits - $140.

16K assembled and tested: $644.

Delivery: Off the shelf to 30 days.

*OFF THE SHELF

Expand your memory. Hurry, use coupon below to order. Or call (609) 921-0321

TECHNICAL DESIGN LABS
Z80
The Design Leader in μ Processing

TECHNICAL DESIGN LABS, INC.
Research Park • Building H
1101 State Rd. • Princeton, N.J. 08540

Please send your Z16 MEMORY FROM THE FUTURE.

KIT: ☐ 4K - $169 ☐ 8K - $295 ☐ 12K - $435
☐ 16K - $574 ☐ 4K Expansion Kit - $140 ☐ Assembled & Tested - $644

Name _____ Street _____
City _____ State _____ Zip _____
I enclose ☐ Check ☐ Money Order - amount of $ _____
Charge Card Data: ☐ BankAmericard or ☐ Master Charge. # _____
Exp. Date _____ Signature _____
☐ Send COD. I enclose 25% deposit. ☐ Send your FREE CATALOG.

TECHNICAL DESIGN LABS INC. • Research Park • Bldg. H • 1101 State Road • Princeton, N.J. 08540

See SWTPC computer equipment at your local dealer

ARIZONA

Byte Shop of Arizona
813 N. Scottsdale Rd.
Temple, Az. 85282

ARKANSAS

Westark Computer Systems, Inc.
2803 Rogers Ave.
Fort Smith, Ark. 72901

CALIFORNIA

Byte Shop of Berkeley
1514 University Ave.
Berkeley, Ca. 94703

Sunshine Computer Co.
9 Palomino Lane
Carson, Calif. 90745

The Electric Brain Computer Store
700 Village Parkway, Suite L
Dublin, Ca. 94566

Computerware
830 First St.
Encinitas, Ca. 92024

Cyberdux
Microcomputer Applications
1210 Santa Fe Dr.
Encinitas, Ca. 92024

A-VID Electronics Co.
1655 East 28th St.
Long Beach, Ca. 90806

The Byte Shop Computer Store #1
1063 El Camino Real
Mountain View, Ca. 94040

Byte Shop of Palo Alto
2227 El Camino
Palo Alto, Ca. 94306

The Computer Center
8205 Ronson Rd.
San Diego, Ca. 92111

Computer Shack
14860 Wicks Blvd.
San Leandro, Ca. 94577

Computer Store of San Francisco
1093 Mission St.
San Francisco, Ca. 94103

Byte Shop
155 Blossom Hill Rd.
San Jose, Ca. 95123

The Byte Shop Computer Store #2
3400 El Camino Real
Santa Clara, Ca. 95051

The Computer Store
820 Broadway
Santa Monica, Ca. 90401

CONNECTICUTT

JRV Computer Store
3714 Whitney Ave.
Hamden, Conn. 06518

FLORIDA

Sunny Computer Stores, Inc.
University Shopping Center
1238 A South Dixie Hwy.
Coral Gables, Fla. 33146

Electronics for Yachting, Inc.
1525 S.E. 16th St.
Ft. Lauderdale, Fl. 33315

Douglas Computer System
Jacksonville, Fl.

Computer Assoc., Inc.
6900 N. Kendall Dr., Suite A103
Miami, Fl. 33156

Microcomputer Systems, Inc.
144 S. Dale Mabry Ave.
Tampa, Fl. 33609

GEORGIA

Atlanta Computer Mart
5091-B Buford Highway
Atlanta, Ga. 30340

ILLINOIS

Semiconductor Specialists
MPV Shop
195 Spangler Ave.
Elmhurst, Ill. 60126

American Microprocessors
Equipment & Supply Corp.
20 N. Milwaukee
Prairie View, Il. 60069

Litlipute Computer Mart, Inc.
4446 Oakton St.
Skokie, Ill. 60076

INDIANA

Data Domain
111 S. College Ave.
Bloomington, In. 47401

Computer Specialties
107 N. Chauncey
W. Lafayette, In. 47906

Syscon International, Inc.
1239 South Bend Ave.
South Bend, In. 46617

KANSAS

Midwest Scientific Instruments, Inc.
220 W. Cedar
Olathe, Ks. 66061

Computer Hut
521 N. Hillside
Wichita, Kansas

KENTUCKY

Cybertronics
312 Production Ct.
Louisville, Ky. 40299

LOUISIANA

Baxter's T.V.
7964 Jefferson Hwy.
Baton Rouge, La. 70809

MARYLAND

The Computer Workshop, Inc.
5709 Fredrick Ave.
Rockville, Md. 20852

MASSACHUSETTS

Computer Warehouse Store
584 Commonwealth Ave.
Boston, Ma. 02215

Computer Mart of Massachusetts
1087 Lexington St.
Waltham, Ma. 02154

MICHIGAN

Mini-Mac, Inc.
303 S. Lemen
Fenton, Mich 48430

MONTANA

Montana Computer Co.
2512 Grand Ave.
Billings, Montana 59102

MINNESOTA

Landers Electric, Inc.
626 S. Holcombe
Litchfield, Minn. 55355

MISSOURI

Computer Workshop of Kansas City
6903 Blair Rd.
Kansas City, Mo. 64152

NEVADA

Johnson T.V. Micro-Computer
2607 E. Charleston
Las Vegas, Nv. 98110

NEW JERSEY

William Electronics Supply
1863 Woodbridge Ave.
Edison, N.J. 08817

Midwest Enterprises, Inc.
815 Standish Ave.
Westfield, N.J. 07090

NEW YORK

Computers Plus
7 Westchester Plaza
Elmsford, N.Y. 10523

Audio Design Electronics
487 Broadway, Room 512
New York, N.Y. 10013

Computer Mart of New York, Inc.
314 Fifth Ave.
New York, N.Y. 10001

OHIO

ELS Systems
2209 N. Taylor Rd.
Cleveland Heights, Oh. 44112

OKLAHOMA

High Technology
1020 Wilshire Blvd.
Oklahoma City, Okla. 73114

OREGON

Byte Shop
2033 S.W. 4th Ave.
Portland, Ore. 97201

PENNSYLVANIA

Martin J. O'Boyle & Assoc.
P.O. Box 9094
Pittsburgh, Pa. 15224

UTAH

The Computer Room
1455 South 1100 East
Salt Lake City, Ut. 84105

TEXAS

Microtex, Inc.
9305-D Harwin Dr.
Houston, Tx. 77036

Electronic Module of Odessa
606 West 10th St.
Odessa, Tx. 79763

The Micro Store
634 S. Central Expressway
Richardson, Tx. 75080

WASHINGTON

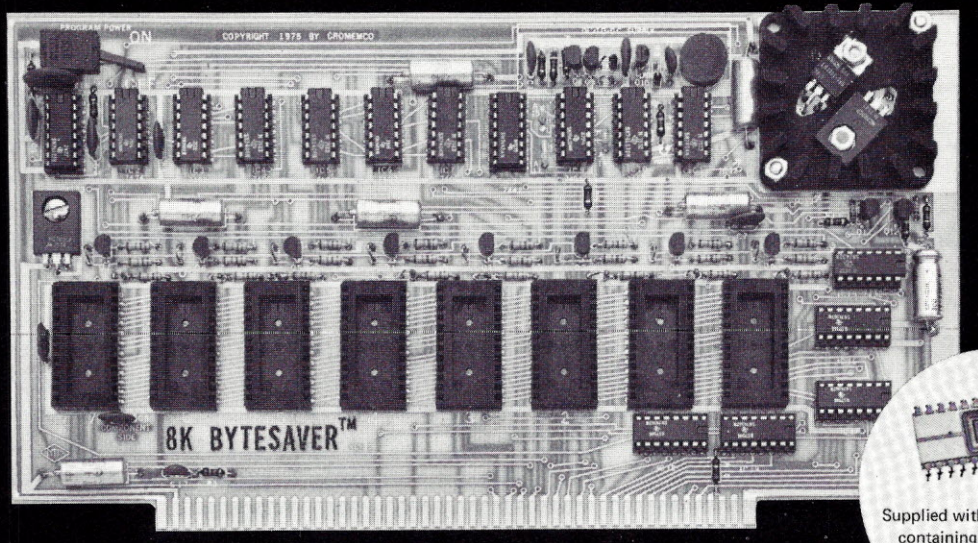
The Retail Computer Store
410 N.E. 72nd St.
Seattle, Wa. 98115

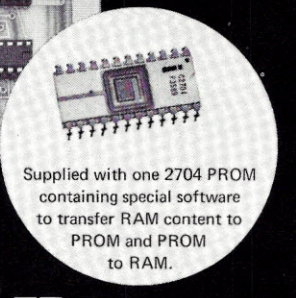
WISCONSIN

The Milwaukee Computer Store
6916 W. North Ave.
Milwaukee, Wi. 53213

WYOMING

Computer Radio Workshop
120 El Dorado Ct.
Cheyenne, Wy. 82001





Supplied with one 2704 PROM containing special software to transfer RAM content to PROM and PROM to RAM.

How to save your programs -- and have a PROM programmer, too

Cromemco's popular BYTESAVER™ memory board gives you two of the most-wanted features in microcomputer work:

(1) a simple, easy way to store your computer programs in programmable read only memory (PROM).

(2) a PROM memory board with the capacity for a full 8K bytes of PROM memory storage.

ECONOMICAL

The BYTESAVER™ is both a place and a way to store programs economically. It transfers programs from the non-permanent computer RAM memory to the permanent PROM memory in the BYTESAVER™ . Once your program is in the BYTESAVER™ , it's protected from power turn-offs, intentional or accidental. The PROMs used with BYTESAVER™ are UV erasable and can be used again and again.

The BYTESAVER™ itself plugs directly into your Altair 8800 or IMSAI 8080.

PROM PROGRAMMER

Many people are surprised to learn that in the BYTESAVER™ you also have your own PROM programmer. But it's so. And it saves you up to hundreds of dollars, since you no longer need to buy one separately.

The built-in programmer is designed for the 2704 and 2708 PROMs. The 2708 holds 1K bytes, four times the capacity of the well-known older 1702 PROM (yet cost-per-byte is about the same). The 2708 is also fast – it lets your computer work at its speed without a wait state. And it's low-powered. With 2708's in all 8 sockets, the BYTESAVER™ is still within MITS bus specifications, drawing only about 500 mA from the +8V bus. A complement of 2708 PROMs gives the BYTESAVER™ its full 8K capacity.

HOLDS LARGE PROGRAMS

The BYTESAVER's™ 8K-byte capacity lets you store the larger and more powerful programs. 8K BASIC, for example, easily fits in the BYTESAVER™ capacity of 8 PROMs. One 1K PROM will hold many games such as Cromemco's DAZZLER-LIFE or DAZZLE-WRITER.

NO KEYBOARD NEEDED

The BYTESAVER™ comes with special software programmed into a 2704 PROM. This software controls transfer of the computer RAM content to the BYTESAVER™ PROM.

So you are ready to go. You don't even need a keyboard. Just set the computer sense switches as instructed in the BYTESAVER™ documentation.

Transfer of memory content to PROM ("burning") takes less than a minute. The BYTESAVER™ software controls computer lights to verify complete and accurate transfer of memory content.

The software also programs any of the other 7 PROM positions in the BYTESAVER™ as readily as the first.

And when used to transfer information from the BYTESAVER™ PROMs to RAM, the special design of the software allows loading a large program such as 8K BASIC in one second.

AVAILABLE NOW – STORE/MAIL

The BYTESAVER™ is sold at computer stores from coast to coast. Or order by mail from Cromemco. Cromemco ships promptly. You can have the BYTESAVER™ in your computer within a week after your order is received.

BYTESAVER™ kit $195
(Model 8KBS-K)

BYTESAVER™ assembled $295
(Model 8KBS-W)

Shipped prepaid if fully paid with order.
California users add 6% sales tax.
Mastercharge and BankAmericard accepted with signed order.

Cromemco
Specialists in computer peripherals
2432 Charleston Rd., Mountain View, CA 94043 • (415) 964-7400

The Directory

COVER THEME: Modular Program Design is represented by red and white blocks, presided over by the control module. This powerful approach to program design is detailed in the article beginning on p. 8. To the lower left is the four board Digital Group System shown plugged into the Standard Mother Board. This review begins on p. 35.

IMPORTANT NOTICE TO SUBSCRIBERS ON PAGE 29 (also see Publisher's Memo, P. 6)

POSTAL NOTICE

MICROTREK, January/February, Volume 1, Number 2. Published bimonthly at 615 Guaranty Building, Cedar Rapids, Iowa 52401. Annual subscription rate for U.S., $10; Canada, $11; all other countries, $12. Second class postage application pending at Cedar Rapids, Iowa 52401.

Forms 3579 and all circulation correspondence should be sent to the address listed above. Please include an address label from a recent issue if possible.

MEMO FROM THE PUBLISHER

PERSONAL COMPUTING is a new and very successful publication in the small computer field. It is published by the Benwill Publishing Corporation of Brookline, Mass. Benwill is also known for DIGITAL DESIGN and other professional electronic publications. Effective with this issue, MICROTREK will have merged with PERSONAL COMPUTING. This simply means that articles and editorial which would normally be published and sent to you with the MICROTREK logo on the cover will now appear as a section in PERSONAL COMPUTING which, as a subscriber to MICROTREK, you will automatically begin to receive. The merger offers many important and immediate benefits to our subscribers.

First, Benwill is an established publishing house with a reputation for getting their publications out on time. Benwill has long since mastered the tedious art of magazine production which seemed to defy our small MICROTREK staff with maddening persistence. Other improvements will appear in the quality, quantity, and graphic presentation of the articles. Full details of the merger appear on page 29 in this issue. Please note that PERSONAL COMPUTING is published bimonthly. However, with the conversion detailed on page 29, you receive more magazine (by page count) for your subscription dollar.

STAFF

EDITOR
Wes Schneider

DESIGN & ART PRODUCTION
Bernadette Moore

PRODUCTION ASSISTANT
Margaret I. Glass

ADVERTISING MANAGER
Stephen Pettus

COVER DESIGN
McKibbon, Douglas, and Richards Advertising

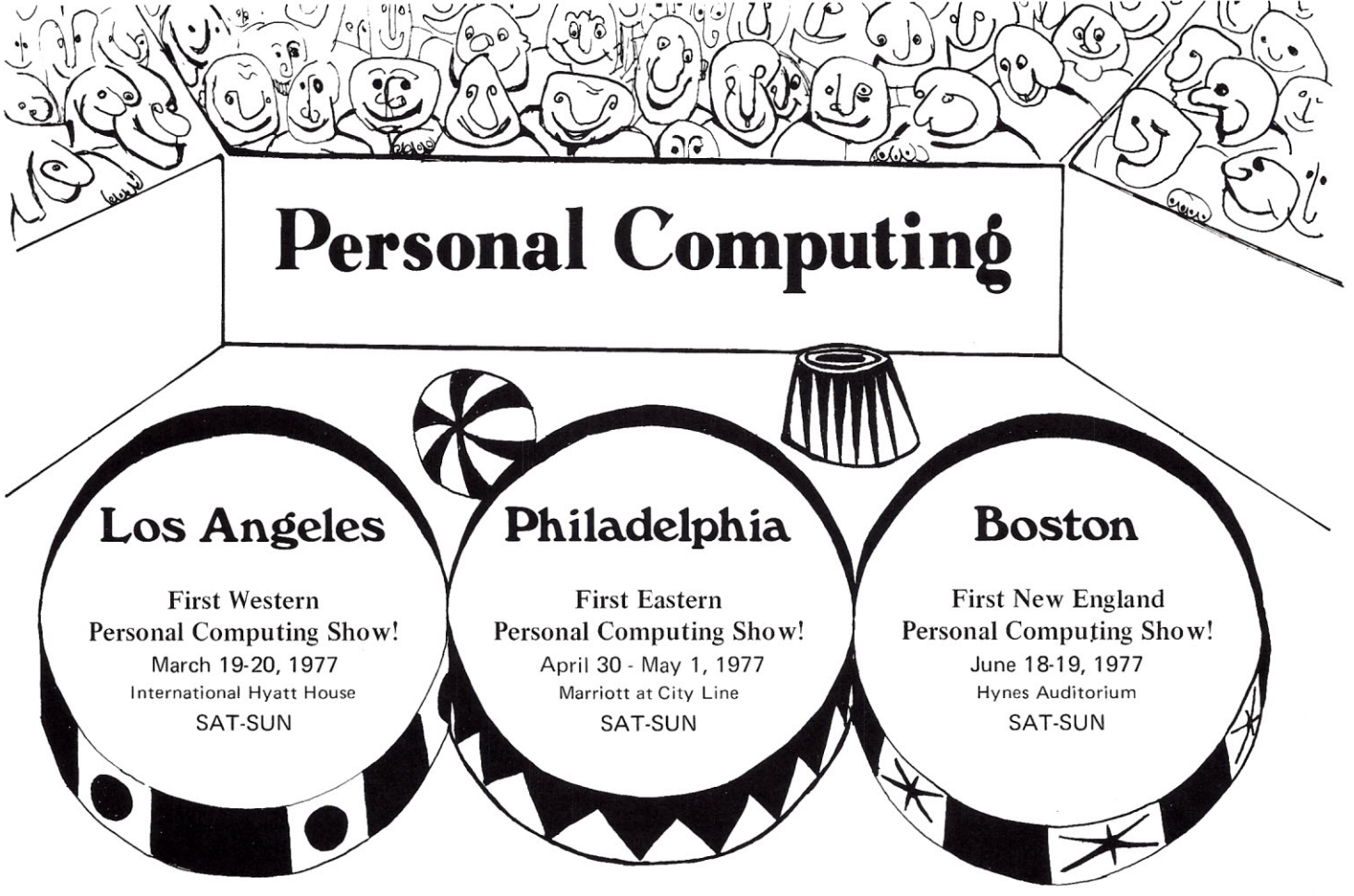
TYPOGRAPHY
Compositors
Compo-Arts

COVER PHOTOGRAPHY
Bob French Studios

PRINTING
Pepco Litho,
Cedar Rapids, Ia.

PUBLISHER
W. Schneider

CIRCULATION
Jonathan Blake



Personal Computing

Los Angeles

First Western
Personal Computing Show!
March 19-20, 1977
International Hyatt House
SAT-SUN

Philadelphia

First Eastern
Personal Computing Show!
April 30 - May 1, 1977
Marriott at City Line
SAT-SUN

Boston

First New England
Personal Computing Show!
June 18-19, 1977
Hynes Auditorium
SAT-SUN

Greatest Computer Shows Ever!

Personal Computing magazine is proud to announce that it is sponsoring the first series of regional Personal Computing Shows.

Beginning with the *Western Personal Computing Show* in Los Angeles, and followed by the *Eastern Personal Computing Show* in Philadelphia and the *New England Personal Computing Show* in Boston, **Personal Computing** magazine intends to make everyone aware of low-cost computing.

Other shows are now being planned for the South, Southwest, Canada, and Europe!

Already, invitations have been sent to all the manufacturers in the personal computing field, computer stores, computer clubs and well-known computer experts.

Special areas of the exhibition halls will be set aside for Personal Computing in Education, in the Home, in HAM Radio, and in Small Businesses. These are all first for a computer show.

Seminars and special presentations include: Computer Synthesized Music, HAM Applications, Trends in Microcomputers, Mass Storage Systems, Lemonade Computer Service Companies, The Kitchen Computer, Computers on the Farm, The Small Business System, Software for Fun and Practical Applications, Computer Club Organization, Standards for the Hobbyists, Computer Art, The House Robot, Computer Crime, Software Protection and Future Computing.

In addition, *special tutorial workshops* will cover all aspects of computer hardware, programming in both machine language and higher-level language and applications. Workshops are designed for both beginners and advanced students in the art of personal computing.

We anticipate 150 different exhibits and crowds of up to 10,000 people at each of these shows. Arrangements for the shows are being handled by a professional management company to ensure that everything runs smoothly.

Cost of Registration:

At the door:
- $10 per show (two days)
- $ 6 per One Day Pass

Special Pre-Registration Rates:
- $ 7.50 per show (two days)
- $ 4.00 per One Day Pass

Note: Show tickets and one day passes entitle you to attend all seminars, workshops, exhibits and other events.

Register Now and Save!

Yes, I would like to take advantage of your special, pre-registration rates. I plan to attend the following regional Personal Computing Show(s):

☐ Los Angeles	☐ Philadelphia	☐ Boston
☐ Show (two days)	☐ Show (two days)	☐ Show (two days)
☐ One Day Pass Only	☐ One day pass	☐ One day pass

Enclosed is a check for _____

Name _____

Address _____

City _____ State & Zip _____

Send to: **Personal Computing**, Conference & Exposition Management Co., Box 844, Greenwich, CT 06830.

modular program design

by:
Steve A. Hughes, Joe Celko,
and Elizabeth M. Hughes
5831 Hillside Drive
Doraville, GA 30340

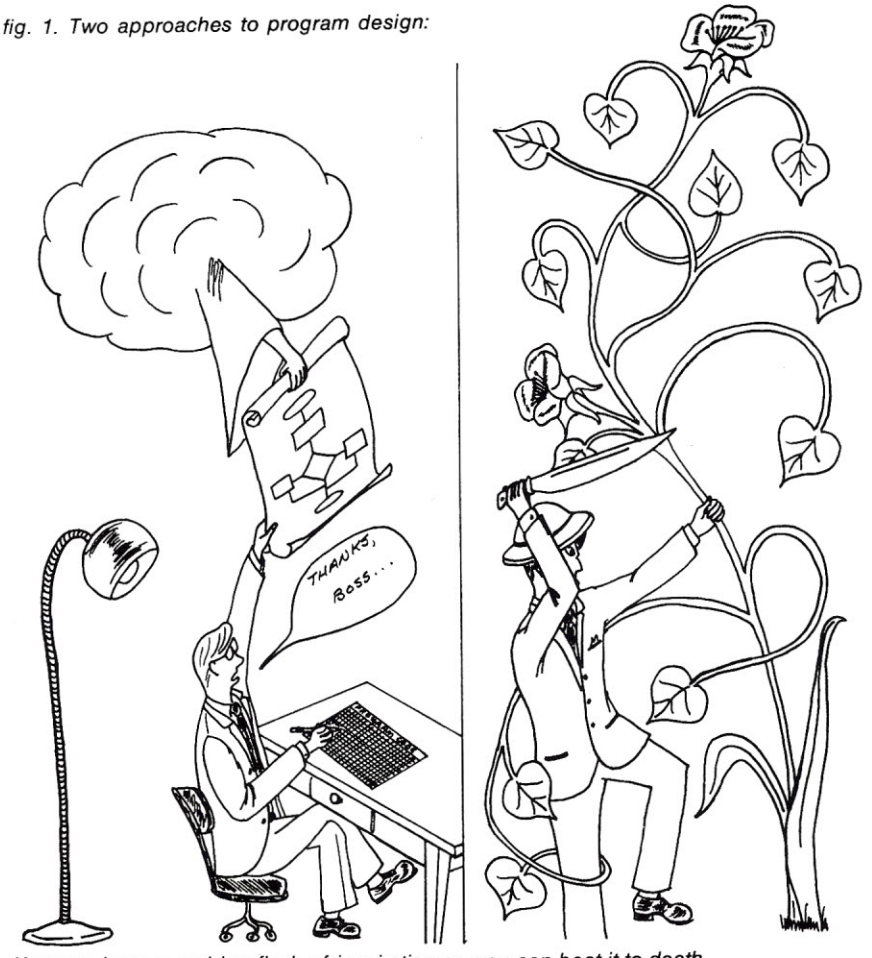
The most difficult phase of programming is program design. For some programmers, arriving at a satisfactory program design is an intuitive process which can neither be readily explained nor taught to anyone who does not share the 'intuition'; but this is not the only, nor is it necessarily the best, approach to developing a program design. The remark of a professor of logic, that "There are two ways of solving a problem — you can have a sudden flash of inspiration or you can beat it to death," is just as applicable to program design as it is to logic problems **(fig. 1).** In fact, a rigorously systematic beat-it-to-death approach will often produce the right answer to a problem more rapidly than the intuitive method, since an intuitive leap to a solution can easily overlook one or more important aspects of the problem.

fear not

In this article we will present a systematic, non-intuitive process for developing workable program designs. Anyone can use it on any programming problem, small or large, simple or complex.

Many programmers of small systems, we have found, are overawed by some of the complex programs they try to create. They take one look at the apparent difficulty of programming a sophisticated game or some other application and decide not to even attempt implementation. Frequently they start writing a program only to end up hopelessly confused by its complexity. Either way, they eventually give up and wait for someone to offer for sale a packaged program which will perform the desired functions. This is unnecessary. By adopting a systematic approach to program design, no programmer need shy away from any problem, regardless of its difficulty. Further, as systematic program design becomes 'second nature', design solutions take less time to develop. This process becomes almost intuitive, yet remaining more reliable than intuition, due to its uniform application to all problems and its certainty of

fig. 1. Two approaches to program design:



You can have a sudden flash of inspiration or you can beat it to death.

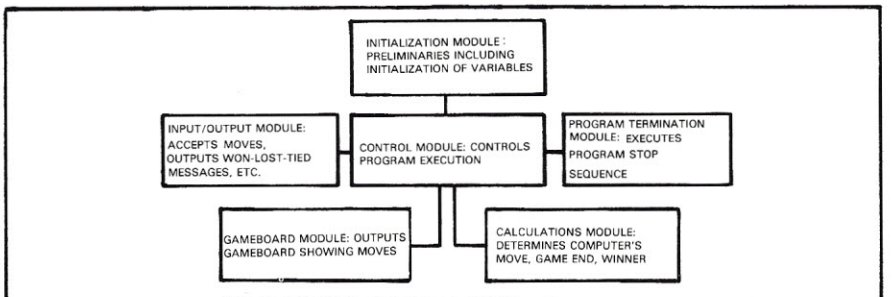


fig. 2. Module chart for a Tic Tac Toe program.

covering all the essentials of the problem at hand.

assessing I/O

The first step in any approach to program design is assessing the problem, deciding exactly what the program must do. One major consideration is the input and output (I/O), particularly in small systems where limited memory may pose a problem if a large amount of data is to be processed.

Let's consider the questions we will need to answer about I/O as an important part of systematic program design. What data will the program require and what will be its form? What information is to be output and what is the most useful format? If formatted output is desired (e.g., a gameboard or tabularized data), rough out the format on paper before attempting the program design. Use this opportunity to optimize the output format, reducing waste from unproductive hard-copy printout (to say nothing of reduced wear and tear on printers). Will there be output which requests input from the operator, such as questions which the operator is to answer while the programming is running? If so, what is the *exact* form of the response which the computer will accept? Whenever a program is interactive, it is wise to indicate to the operator what the specific options and formats are for correct responses to computer interrogations; is "Y" or "YES" the correct response for Yes?

user convenience

A user will always do things the easiest way, and if using the computer program is more trouble than doing the job manually, he simply won't use it. Suppose a programmer writes a grocery-list program. He stores the Universal Product Codes for all available groceries, intending that his wife enter the quantity for each item as needed. The grocery-list program then directs the computer to print out the complete shopping list. Let's run through a hypothetical program run to spot the shortcomings of the program in terms of user convenience. First, each product is stored by its Universal Product Code. How many people know the Code by heart? Next, the program demands of us that if we do not want to order any quantity of a given product, we *must* enter a zero to indicate this. Imagine the time it would take to do this for each product available in the average supermarket! No one in their right mind would use such a system. The ease with which a program can be used will determine to a surprising extent its frequency of use; programs *must* be engineered for user convenience.

initial values

After the I/O format and options have been settled, the next consideration is to list the critical operations which the program is to perform. Making such a list will guarantee that

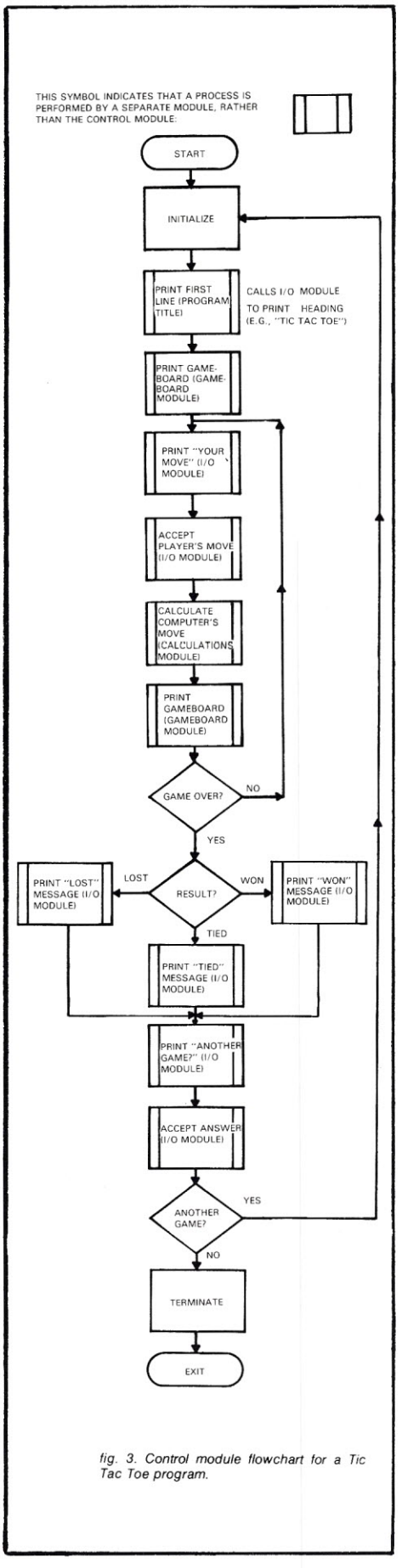


fig. 3. Control module flowchart for a Tic Tac Toe program.

none are forgotten in the development of a complex program.

When the program variables and constants have been determined, initial values should be specified and any other preparations (such as I/O initialization, etc.) listed.

Tic Tac Toe program

To illustrate the process of systematic program design, let's consider how we might approach the design of a program to play Tic Tac Toe. We begin by assessing the operations which the computer is to perform. The program must output the gameboard, accept the player's move, decide what move to make and output the revised gameboard showing both moves. When appropriate, it must recognize that the game is over and declare a winner. Finally, it must stop under program control.

The activities necessary to carry on the game of Tic Tac Toe can be grouped into related tasks. For example, one whole section of the program can be devoted to outputting the gameboard in its original and its revised states. Another task grouping would deal with accepting input and producing any output which is not a part of the game board, such as a prompting message to the player indicating the format of his play responses or the final won/lost/-tied message at the conclusion of the game. A third program section is required for the calculations of the computer's best move and for the 'game over' determination. Formal program termination composes the final section of this or any other program.

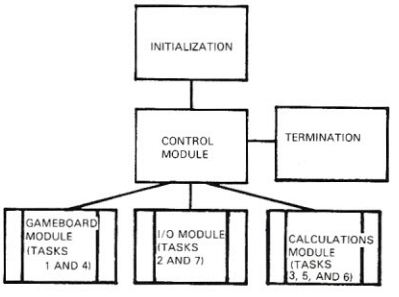
fig. 4. The evolution of a control module.

STAGE 1: SCRATCHLIST OF PROGRAM TASKS

1. PRINT GAMEBOARD
2. ACCEPT PLAYER'S MOVE
3. CALCULATE COMPUTER'S MOVE
4. PRINT REVISED GAMEBOARD
5. RECOGNIZE END OF GAME
6. DETERMINE WINNER
7. PRINT WON/LOST/TIED MESSAGE

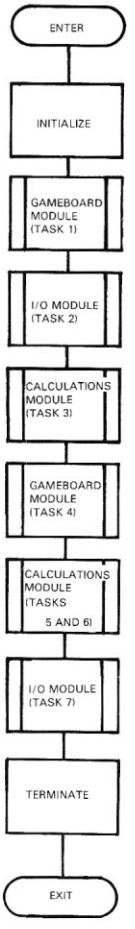
OTHER TASKS CAN BE ADDED, IF DESIRED, SUCH AS LETTING PLAYER PICK WHO HAS FIRST MOVE OR KEEPING SCORE OF GAMES WON/LOST/TIED.

STAGE 2: MODULE CHART



TERMINATION, ESSENTIAL TO EVERY PROGRAM, MAY EITHER BE SHOWN AND HANDLED AS A SEPARATE PROGRAM ELEMENT (SHOWNHERE) OR AS A MODULE, IF PREFERABLE.

STAGE 3: ROUGH CONTROL MODULE FLOWCHART



STAGE 4: DETAILED CONTROL MODULE FLOWCHART IS SHOWN IN FIG. 3

program modules

As we have seen, the Tic Tac Toe program actually consisted of a relatively few task groups. Each of these tasks can be performed by separate sections of code which are brought into use by the main program as needed. The identification of these functional program *modules constitutes an important part of the systematic method for program design. Almost every program includes I/O, calculations, and termination modules. In this particular game example we considered the task of outputting the gameboard separately, since it represents a different kind of I/O problem. This is the best approach in programs which must support an interactive environment which includes a television-type display.*

module chart

The value of listing all the tasks which a program is to perform is that it shows clearly the modules which will be needed to create the program. Reducing all but the simplest programs to small, *workable* sections like these will always make programming go faster and with fewer errors. For the task list to be most useful, however, it should be sketched up into a *module chart,* which is the first (tentative) expression of the program design **(fig. 2).** A module chart identifies each of the functional sections composing the program and introduces the keystone into the program-edifice: the *control module.*

control module

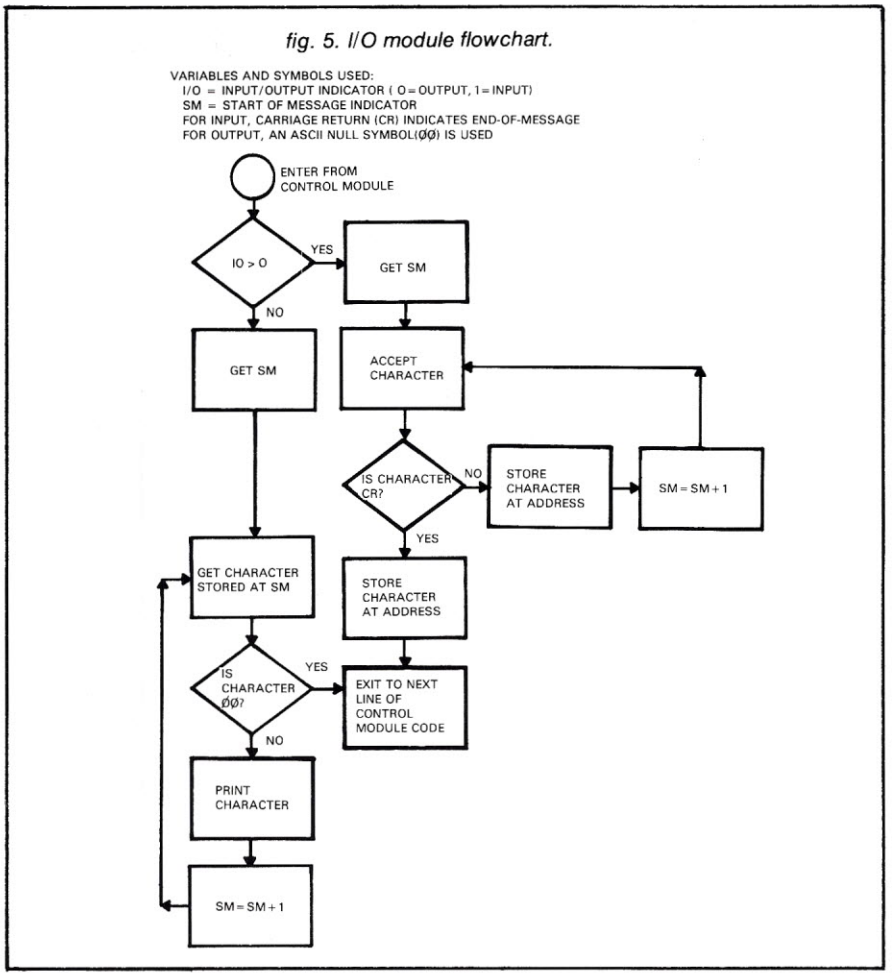
A control module does exactly what one would expect: it controls the program's operation. The control module is responsible for the program's *decisions* and brings the individual modules which perform major program functions, like I/O, into play as they are needed. Perhaps the best (or at least the most amusing) analogy for a control module is the shepherd who kept "tuned sheep". Each sheep had a bell tied around its neck. When the shepherd tapped the sheep on the head with his crook, the sheep would nod its head, ringing its bell. By hitting the sheep in the desired order, the shepherd could play tunes. This is analogous to the operation of a control module. It decides which operation is to be performed next (which note to play) and calls (hits) the appropriate module (sheep). The module (sheep) then performs the required operation (nods its head and rings its bell). This is the essence of the control module function; but since the control module must call all the

other modules as needed, its flowchart closely resembles a macro-flowchart of the program. On a module chart, the control module is shown immediately after the required initialization operations, and all remaining modules are shown attached to it **(fig. 2).**

Fig. 3 shows a flowchart of the control module for the Tic Tac Toe program, indicating the calling sequence to the function modules. This flowchart helps the programmer visualize the relationship between the control module and the other modules more clearly than the module chart, since it indicates at which points in the program's operation each module is called.

The control module is, as we have seen, the program's structure. Consequently, a well designed control module is half the battle in developing a good program. One of the advantages of using a modular approach to program design is that it reduces the apparent complexity of the problem to a minimum, allowing the programmer to concentrate on the essential details. This is particularly evident during the design of the control module, for it is here that the entire program is set up. If the data required for the operation of each module is completely defined, along with the required data formats, and the calling sequence is determined from the control module flowchart, design of the control module becomes a straightforward process. **Fig. 4** illustrates the evolution of a control module from the list of program tasks to the final form.

fig. 5. I/O module flowchart.



I/O module example

When the control module has been flowcharted, the programmer can begin to flowchart the individual function modules. **Fig. 5** shows the generalized I/O module flowchart. This module examines a variable (IO) to determine whether the operation required will be an input or an output. The module then finds the starting address of the message from another variable (SM). If the operation calls for data input to computer memory, the I/O module stores the first character it receives at the Start of Message address and the addresses which follow until it receives the *control character* (CR = carriage return) which indicates the end of the message or the conclusion of data. If, on the other hand, output is requested, the I/O module outputs data stored at the Start of Message address and following addresses until the end-of-message character is reached. Note that in either case, reaching the end-of-message character causes the I/O module to return control to the control module.

modules vs subroutines

In extremely simple programs, individual routines are used only once. A *routine* is simply a block of computer instructions (code) designed to perform a specific task. Thus, a program to average a set of numbers may treat the section which calculates the average as a single block of code; a routine. If however, the programmer wants to find the averages of several sets of numbers, the averaging routine must be repeated for each set of numbers to be averaged. To meet this need, he might employ a subroutine, since subroutines are an efficient method for dealing with code which would otherwise appear repeatedly in the mainstream of the program. A *subroutine* is a section of code which performs a specific task, which can be called into use by any part of the program. Once

The POLY 88 Microcomputer System

The POLY 88 Microcomputer System brings to the user, in one compact package, the capability of developing programs and hardware as well as enjoying the interaction with computers. The POLY 88 System uses a video monitor for display, a keyboard for input, and cassette tape for storage. The system will connect to a hard-copy terminal. POLY 88 hardware consists of CPU circuit card with on-board memory and I/O, video display circuit card with keyboard input port and graphics capability, and mini-cards that connect to the CPU board via ribbon cable for cassette or serial interface.

Central Processor Card features 512 bytes of RAM (random access memory), plus sockets for up to 3K of 2708-type PROM or ROM (read only memory), vectored interrrupt, and real time clock, as well as an optional serial input-output port featuring software-controlled baud rate.

The Video Terminal Interface circuit card will generate 16 lines of 32 or 64 characters or 48 x 128 graphics grid on a standard video monitor or slightly modified TV set. It also includes an 8 bit parallel keyboard input port, allowing the board to function as the complete computer interface.

The POLY 88 Chassis and backplane/motherboard is Altair compatible,and will provide 6 amps of power to five cards. Only two switches are mounted on the front panel — a lighted on/off switch, and a push-button, which, in addition to resetting the system, indicates a halt condition in the processor.

The firmware Monitor is integral to the POLY 88 System. This 1024 byte program in ROM allows the user to display data on a TV screen, enter data into memory using a keyboard, read and dump data to the cassette interface in a standard format, and single step through a program while displaying the contents of each of the 8080's internal registers. The monitor provides many of the standard driving routines to greatly simplify the job of programming. The monitor forms the basis of an ever-increasing software library available to POLY 88 owners.

Options from PolyMorphic Systems include a board with 8K of additional RAM, a versatile prototyping board ("The Ideaboard"), and an Analog Interface. Also, many "Altair-compatible" products on the market are compatible with this system.

Prices: Basic kit including chassis, CPU and video cards — $595, $795 assembled. Cassette option — $90 kit and $125 assembled. 8K of RAM — $300 in kit form or $385 assembled. We also sell the video and other "Altair-compatible" circuit cards separately.

Dealers: This system sells itself.

All prices and specifications subject to change without notice. Prices are USA only. California residents add 6% sales tax.

Prepaid orders shipped postpaid. BankAmericard and Master Charge accepted.

☐ Please send more information

☐ Order and check enclosed

Name _____

Address _____

City/State _____

Zip _____

BankAmericard _____ Expires ____

Master Charge _____ Expires ____

Signature _____

PolyMorphic Systems (805) 967-2351
737 South Kellogg Avenue Goleta, CA 93017

PolyMorphic Systems

its task is complete, the subroutine returns control to the point in the main program from which it was called. Subroutines are designed to work on any variables they receive. This is an important point, as we shall see, and one which gives rise to the most infuriating bugs ever to plague programmers. If the wrong variable is somehow passed to the subroutine (arguments passed to subroutines are called "parameters"), the variable's value will change without the programmer's being aware of it. This error will most likely ripple through the entire program, causing further problems in the program's operation which can continue undetected until the programmer finds that his output is gibberish. The process of isolating and neutralizing this sort of bug is generally long and troublesome.

A module, on the other hand, is designed to avoid these difficulties from the start. Like a subroutine, a module is a body of code which performs a single task; and, like some subroutines, it has only one point of entry and only one exit. This means that whenever a module is employed, it starts at a well defined beginning, goes through a particular process to accomplish its task, and stops its activity at a single point. There is, however, an essential difference between modules and subroutines. The *only* variables which a module can change are those which it has been designed to use for receiving data from the control module and for making information available for other modules to use. This restriction completely rules out the possibility of its changing any *common* variable (i.e., one which can be accessed and used by more than one module) which is not *specifically* designated for its use. Within its own structure a module may perform operations on internal

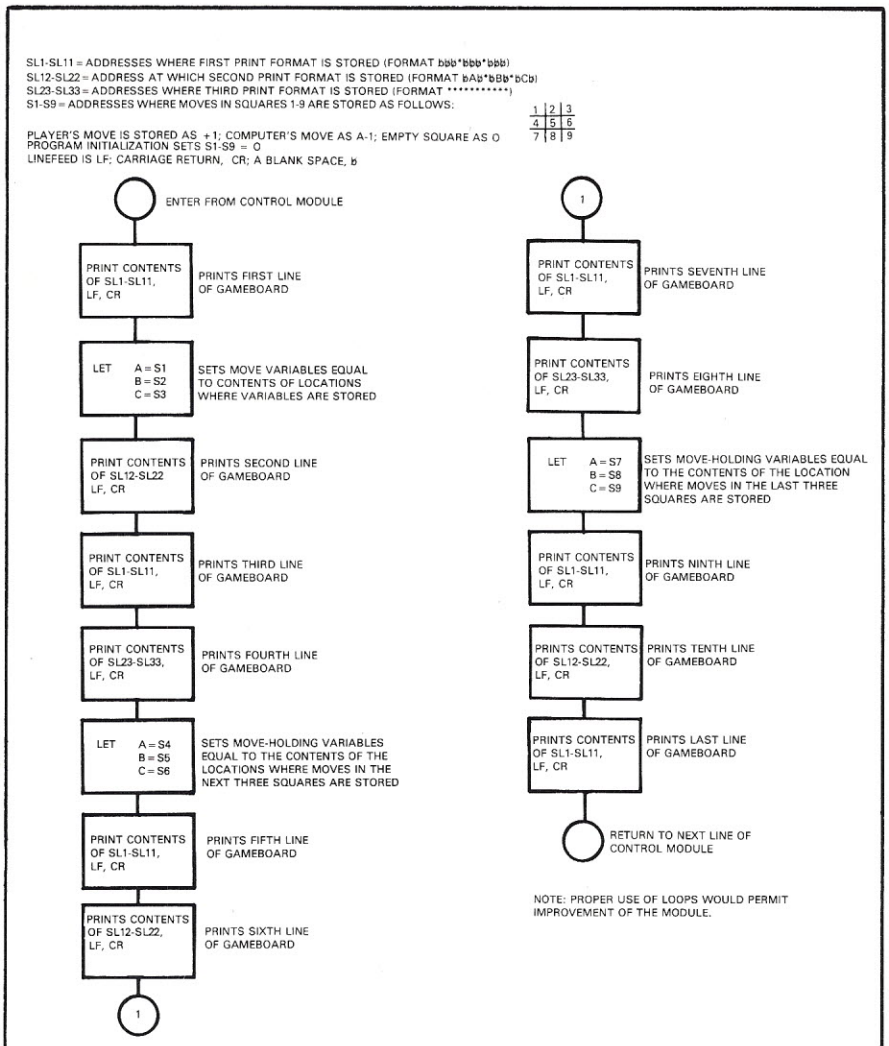


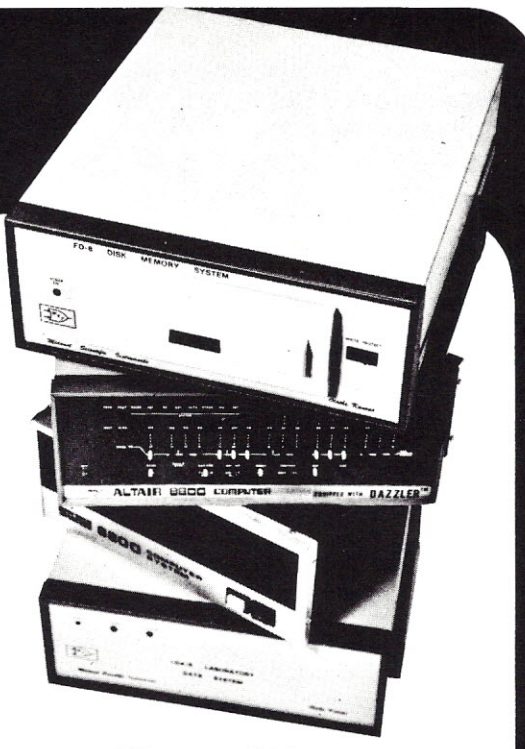
fig. 6. Gameboard module.

variables (called *local* variables), but since these variables will never appear outside that particular module, no mysterious catastrophic failures can occur.

linking variables

Returning to **fig. 5,** let's see how the I/O module operates according to our concept of modularity. First, the I/O module accepts and uses the variable called I/O, set by the control module, to decide whether it is to perform an input or an output operation; but it doesn't change the value of IO, and IO is only used by the I/O module and the control module. These two variables provide the sole link between the I/O module and the rest of the program. With only these two links, there is little possibility of the I/O module spreading errors through the rest of the program. That, in a nutshell, is the advantage of modular programming. It also points out the care which must be taken in preparing the control module to assure that all common or "global" variables are in a form acceptable to any modules which use them. This is rarely difficult, however, since frequently only the control module and one other functional module will share some common variable. Also, the care required in establishing common variables is counterbalanced by

No Matter How You Stack Them Ours Comes Out On Top!



OUR FD-8 FLOPPY DISK SYSTEM interfaces to anybody's microcomputer system via a single PIA chip!

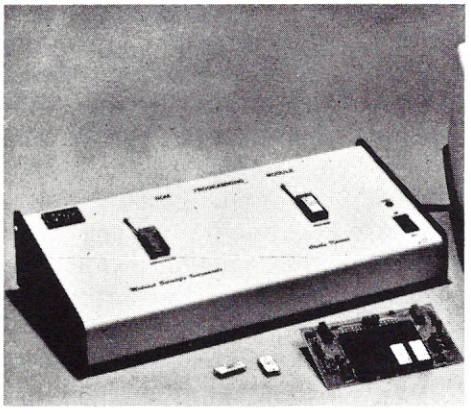
If you don't use PIA's, then one 8 bit bidirectional data port and one output only control port is all that's required—that simple! If you don't have a bidirectional port then separate input and output ports will do.

Full sector buffering in 3K of RAM contained on the controller card itself eliminates any dependence upon processor speed.

Each drive is contained in its own cabinet with power supply. Up to 4 drives may be daisy chained together and selected under software control from a single controller card.

Both single and double density, single or multiple drive units are available.

Complete F DOS software for both 8080 and 6800 systems is provided at no additional charge, including: disk driver subroutines, variable length file management system, disk assembler/editor, and integration with basic.



MSI introduces the PR-1 PROM Programmer and verification module for use with microprocessor systems.

The PR-1 interfaces to any microcomputer system via a single PIA chip. The unit is designed to program 1702A PROMS. Complete software for PROM programming is provided with the system at no additional charge.

MSI software products including our mini assembler, disassembler, and basic are now available on KC standard cassettes. Please specify either paper tape or cassette when ordering.

For the past 6 years MSI has been a leading manufacturer of microcomputer data handling systems for medical laboratories, so we're not new to the business. Four years ago we introduced the first floppy disk system for programmable calculators, which is still in production today. We manufacture CRT terminals, PROM programmers, and a large selection of instrumentation interfaces.
For more comprehensive product information, write MSI at the address below. Incidentally, our products are ready for immediate delivery.

Master Charge & BankAmericard orders welcome.

Here are two MSI Dealers, who can show you our products in action...

Microcomputer Systems, Inc.
144 S. Dale Mabry Ave.
Tampa, Florida 33609
(813) 879-4301

American Microprocessors,
Equipment & Supply Corp.
Chicagoland Airport, P.O. Box 515
Prairie View, Illinois 60069
(312) 634-0076

Midwest Scientific Instruments

220 WEST CEDAR, OLATHE, KANSAS 66061 • PHONE 913 764-3273 • TWX 910 749 6403 (MSI OLAT)

the ease of establishing local variables, since a local variable only needs to be in a form suitable to the module which is its locale.

portable modules

Once modules are fully understood, the process of developing the remainder of the program becomes simple. After flowcharting the control module, the remaining modules should be flowcharted, especially if their operations are at all complex. The main consideration in flowcharting individual modules is to make them highly *general* from the standpoint of applications. This is done partially to avoid limiting the program's applicability in unnecessary ways and partially to make the module as portable as possible. A portable module can be moved from one program to another with almost no alterations if it is sufficiently general. This is exemplified by the I/O module discussed above, which can be used anywhere that an input/output module is required with few, if any, modifications.

modular modification

Two modules in the Tic Tac Toe program are tied to this particular game: the gameboard output module and the calculations module. **Fig. 6** shows the flowchart for the gameboard module. Note that only three different line-output formats are used to output the entire board, including the moves by both players at the various stages of the game. Although it is unlikely that many uses could be found for this module apart from the Tic Tac Toe game, it is easy to see how simple a matter it would be to modify it, if desired. This is yet another advantage of modular programming. Instead of requiring that the entire program be rewritten whenever an improvement or alteration is desirable, only the few modules which affect the change need be modified and re-debugged. This means, for example, that should a programmer wish to modify his Tic Tac Toe program to permit play on a 4 X 4 board, he only need rewrite the gameboard and the calculations modules in order to implement the change. Further, since each module can be checked out independently of the other modules, the programmer could continue to use the original 3 X 3 game until he was ready to replace the two modules with their 4 X 4 versions.

calculations module

The calculations module, shown in **fig. 7,** is the one exception to the rule that modules should be generalized for portability; but since the main distinguishing feature of a program is its set of calculations, it is rarely either possible or desirable to move a calculations module from one program to another. It frequently occurs, however, that programmers discover improved ways of performing the calculations for programs which are up and running. When this is the case, a modular approach to a program's calculations earns its keep. The improvement can be incorporated into a new module, tested (by observing its output while the input is varied over the range of acceptable input values, including boundary values), and simply plugged into the original program which employed the former calculations module, without numerous bugs and similar problems commonly associated with an analogous substitution in a non-modular program. Whether the change is an improved calculations technique or an improvement in the game (e.g., a larger gameboard or the addition of a third dimension), the modular structure makes it easy to implement.

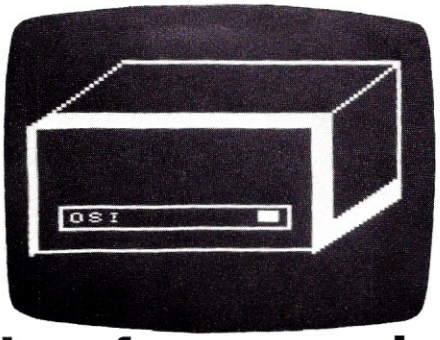
one step at a time

After each module has been flowcharted, it can be coded and tested independently of the rest of the program. Since, in order to write a module, it is necessary to know the form in which data is used by it, one can easily write dummy code which gives it test data in the correct form and outputs what the module would produce if it were connected to the rest of the program. Similarly, the control module can be tested by inserting dummy code at the points where it would call other modules (often a simple RETURN instruction will suffice). The remaining modules can be tested separately before being added to the program; or, after the control module has been debugged, they can be tested, one-by-one, from their actual positions in the program. In this way, the program can be built up in piecemeal fashion, with no more than one section of it needing debugging at any given time. This simplifies the debugging process, permitting the programmer to get his system up and running sooner than if the whole program had to be debugged simultaneously.

criticisms

Two disadvantages are occasionally advanced against modular programming, namely, that it takes more storage to run a modular than a non-modular program, and that it takes more time to write a modular program than a non-modular one. Both of these criticisms are misleading. If the time required for writing a program includes checkout and debugging time, modular programming is almost always much faster. It is easy to overlook debugging time until one actually has to debug a program, but the dramatic reductions in checkout time provided by modular program design more than compensates for any slight increase in the time re-

Meet the Challenger.™



The Challenger Self Portrait

The new price and performance champ from OSI.

He's got his act together!
Even our lowest-cost Challenger comes fully assembled, complete with a 500 ns 6502A, serial interface, 1,024 words of memory and a UL-approved power supply, all for $439. Every Challenger comes ready for easy expansion with an 8-slot mother board, backplane expansion capability, and a power supply heavy enough to handle a full complement of system boards. Our 4K Challenger comes ready to run BASIC minutes after you unpack it. And there's more.

He packs some heavy hardware.
You've never seen memory and interface options like these—not at our prices, fully assembled! 4K RAM memory boards $139! (see below). Single drive OSI Challenger Floppy Disk $990! Dual drive Floppy $1490! Plus 8K PROM boards! A Video Graphics board, including alphabetics, graphics, and **color!** An audio cassette, A/D, D/A and parallel I/O board! A backplane extender board! A prototyping board! And our extraordinary **CPU Expander Board**—it lets you run a Z-80, and 6100 (PDP-8 equivalent) **concurrently** with The Challenger's 6502, or under its control.

There's nothing soft about his software!
OSI has full software support for our Challengers. Including extended BASIC, extended Video Monitor, a Disk Operating System, some very Hollywood real time programs for Video Graphics, Animation, Sound Processing and so forth, plus PROM firmware, with more to come.

He's fast!
You can order The Challenger with a 6502C for a 250 ns cycle time, with a standard 6502A for 500 ns cycle time, or with a 6800 for 1 microsecond cycle time. And with our CPU Expander Board, you can always update to any new CPU to be as fast as fast can be.

And he isn't just good!
He's better! By design. The OSI Challenger is the only completely-assembled, ultra-high-performance, fully-expandable mainframe computer that does this much for this little. Get your hands on one now. Send for your Challenger today.

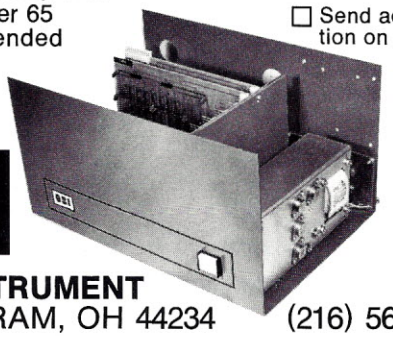
You can't beat The Challenger!
The OSI Challenger 65-1K. Fully assembled. Features 6502A CPU, serial interface, 1,024 words of memory. **$439.**
The OSI Challenger 65-4K. Same as 65-1K but with 4,096 words of memory. Will run Tiny BASIC without expansion. **$529.**
The OSI Challenger 65V-4K. NO NEED for an expensive terminal. Connects to your ASCII keyboard and video monitor through included OSI 440 Video Board. Features software utility that simulates a deluxe CRT terminal. **$675.**
The OSI Challenger 68-1K. Based on 6800 CPU. For the casual hobbyist, smaller systems. The Challenger 68 series comes only in serial interface forms and is compatible with MIKBug software through an included OSI software utilities package. **$459.**
The OSI Challenger 68-4K. With OSI 4K BASIC on paper tape. **$529**
SPECIAL! ADDITIONAL 4K MEMORY BOARDS. Ordered with your Challenger, limit 3 more at this special Low Price, (total 16K, including 4K already on-board in mainframe). **$139**
Buy 12K or larger Challenger 65 system and we include Extended BASIC FREE!
OSI Challenger Floppy Disk System. Fully assembled, for use with OSI Computers only. **$990** Single drive **$1490** Dual drive.
OSI Audio Cassette Interface. Comes assembled, but with room for you to populate with A/D and D/A chips later. (OSI 430 based) **$89**
And all the baseboards and kits of the powerful OSI 400 System.

OK, OSI, I'm ready to buy!
To order your Challenger System, send the total amount of your purchase plus $4.00 for shipping and insurance (plus sales tax for Ohio orders) by personal money order or check. Or indicate **all** numbers on your BankAmericard or Master Charge to charge your order.
Or send a 20% (non-refundable) deposit to receive your order C.O.D. Delivery is typically 60 days (except when payment is by check, which must clear before shipment can be made). Deliveries are scheduled on a first ordered, first shipped basis.

Name_____
Address_____
City_____ State_____ Zip_____
Telephone_____
Bank card info Inter Bank #_____
Expiration Date_____
Account # _____

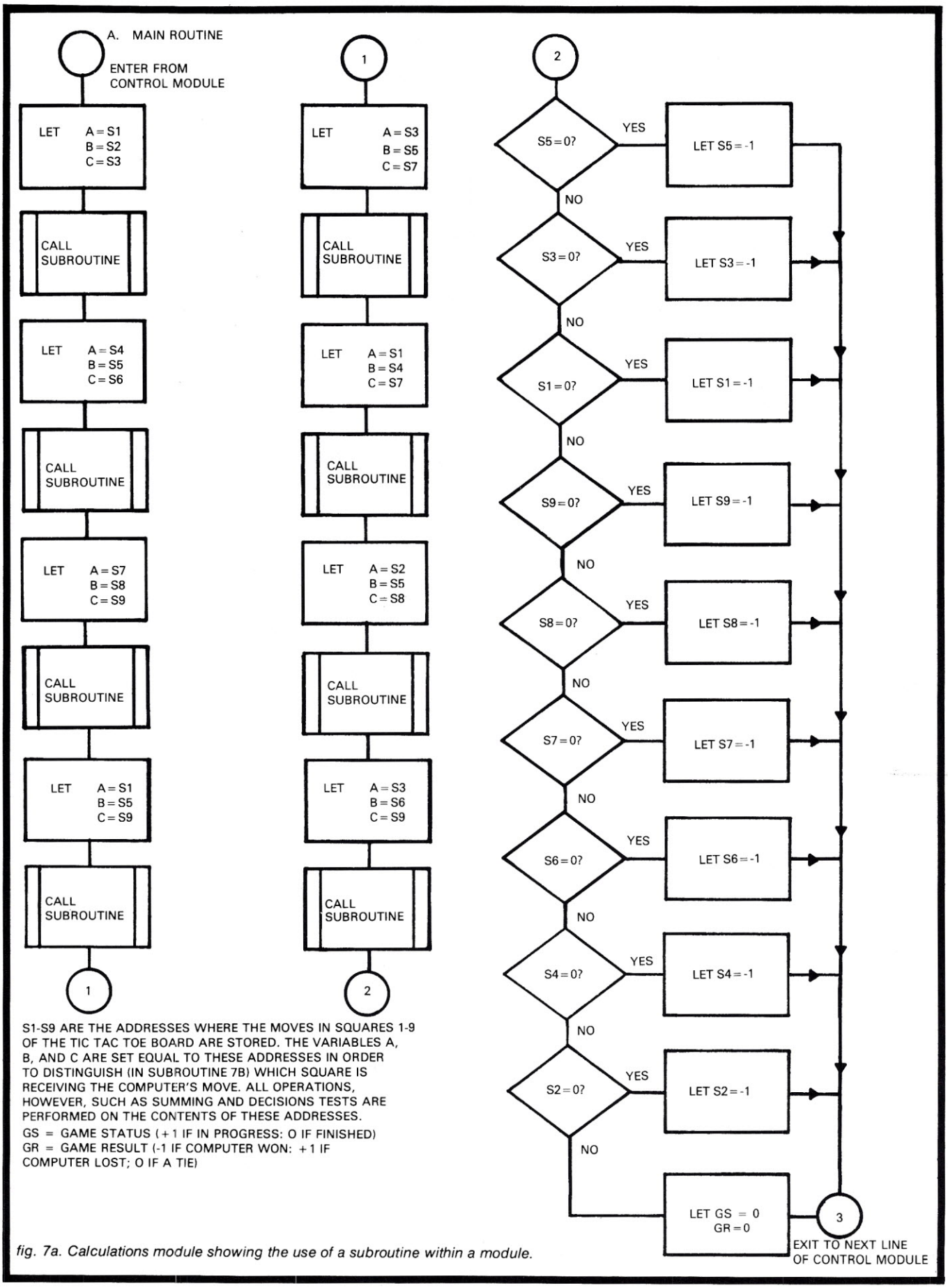
Check ☐ M. O. ☐ BAC ☐ MC ☐
20%, bal. C.O.D. ☐

☐ Order attached.
☐ Send additional information on The OSI Challenger.
☐ Send additional information on OSI 400 Kits.
☐ $1.00 enclosed for complete OSI Catalog.



OSI

OHIO SCIENTIFIC INSTRUMENT
Dept. M 11679 HAYDEN STREET, HIRAM, OH 44234 (216) 569-7945



S1-S9 ARE THE ADDRESSES WHERE THE MOVES IN SQUARES 1-9 OF THE TIC TAC TOE BOARD ARE STORED. THE VARIABLES A, B, AND C ARE SET EQUAL TO THESE ADDRESSES IN ORDER TO DISTINGUISH (IN SUBROUTINE 7B) WHICH SQUARE IS RECEIVING THE COMPUTER'S MOVE. ALL OPERATIONS, HOWEVER, SUCH AS SUMMING AND DECISIONS TESTS ARE PERFORMED ON THE CONTENTS OF THESE ADDRESSES.

GS = GAME STATUS (+1 IF IN PROGRESS: 0 IF FINISHED)
GR = GAME RESULT (-1 IF COMPUTER WON: +1 IF COMPUTER LOST; 0 IF A TIE)

fig. 7a. Calculations module showing the use of a subroutine within a module.

quired to develop the original code. This is nearly always the case when a job is thought out before it is put on paper: the programmer who thinks it through may not start as soon as the one who just charges in, but he usually has less trouble with it and emerges with the right solution first. The complaint about storage is similarly ill-founded. Although the best possible straightline program will usually take less storage space than the best modular program, for a given task, few programmers produce the best possible program for their problem with regularity. Given the normal programming conditions, modular design almost invariably requires less storage. An added benefit of modular programming on systems with limited memory is that, if the whole program won't fit in the available storage at once, a module can be read in when it is needed, then read back out again, along with its local variables (onto tape or a disc memory) when it is not in use. This brings within the range of limited-memory systems programs which simply couldn't be run otherwise, since partial program loading of this sort is a sheer impossibility with a non-modular program.

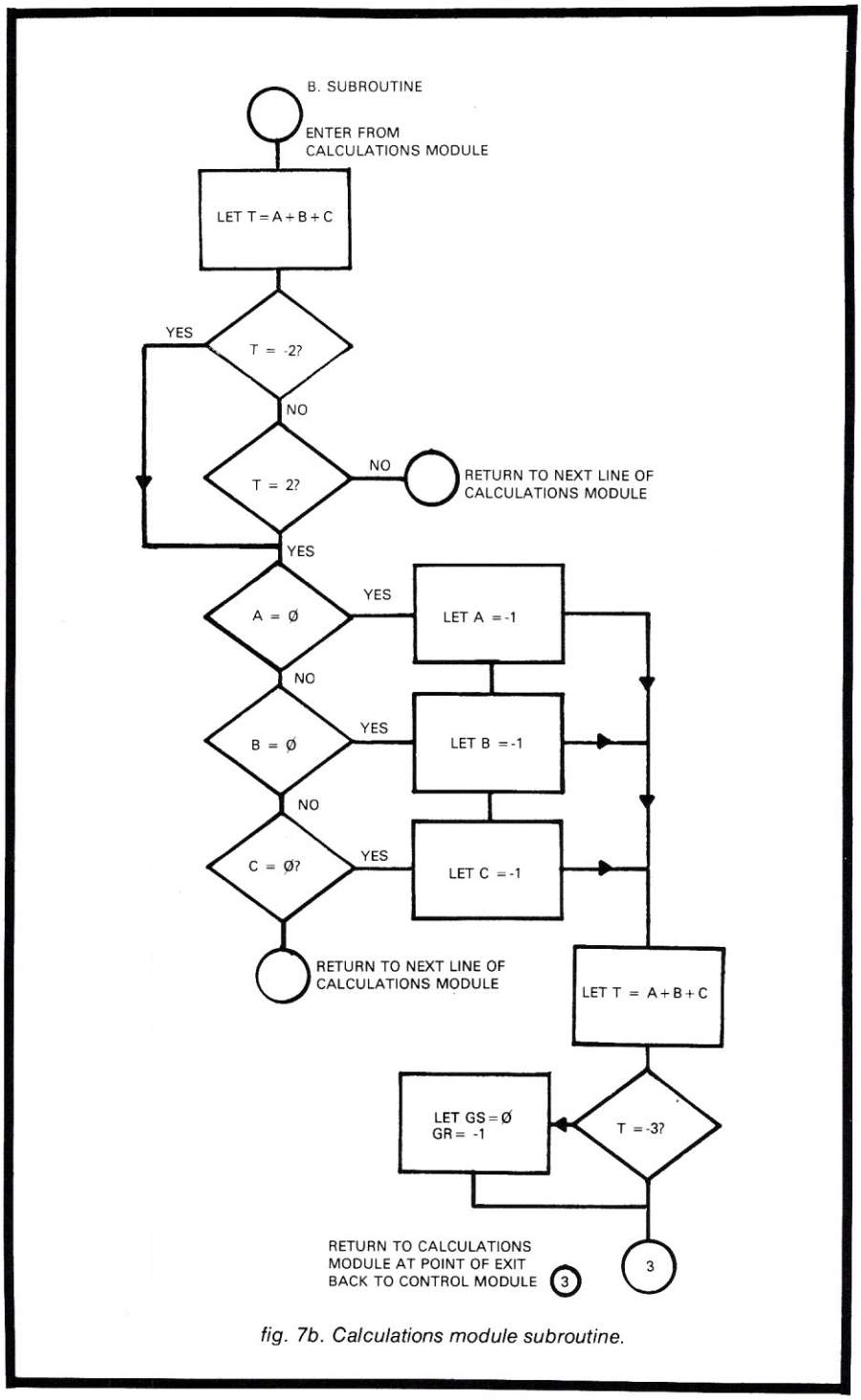


fig. 7b. Calculations module subroutine.

conclusion

The approach we recommend for program design, then, is not only simple and direct, but applicable to any problem which faces the small systems programmer. The program design process can be summarized in a few statements:

1 — Assess the problem

2 — Divide the problem into single tasks

3 — Draw a module chart

4 — Flowchart the control module

5 — Flowchart the remaining modules

6 — Code and test the modules individually, assembling them into the complete program.

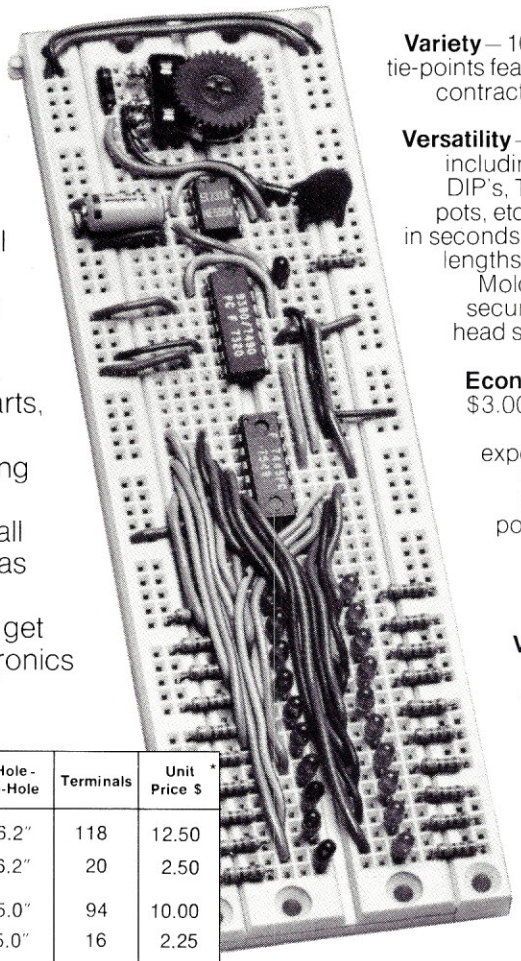
The result: a finished *working* program. Some extremely simple programs are best treated as single modules, but with any program of more than minimal complexity, going through the steps listed above will probably produce the quickest and most reliable solution. The next time you start wishing you could have a program that played some complicated game or handled an involved application, don't just wish — use our modular approach to divide and conquer the problem. Even the most awe-inspiring programs are easy to handle when they're divided into smaller, workable sections — and that's what modular programming is all about. END

EVERY PROJECT IN THIS BOOK IS ANOTHER REASON TO OWN CSC'S QT SOCKETS AND BUS STRIPS.

With QT solderless breadboarding sockets and bus strips, you can build twice the projects in half the time. Because making connections or circuit changes is as fast as pushing in—or pulling out—component leads. No special clips or jumpers required, either.

When you're building circuits just for the fun of it, you can take them apart in minutes—not hours. So you save money by re-using parts, while eliminating heat damage to expensive components. Interlocking QT Sockets and Bus Strips are infinitely expandable, too: start small and "grow" breadboards as large as you wish.

For as little as $3.00, you can get a lot more out of your time in electronics—so why not treat yourself to a QT Socket today?



Variety—10 models from 70 to 590 solderless tie-points feature snap/lock design to expand or contract your breadboard to fit every circuit and budget requirement.

Versatility—Use with virtually all types of parts, including resistors, capacitors, transistors, DIP's, TO-5's, LED's, transformers, relays, pots, etc. Most plug-in directly and instantly, in seconds. No special jumpers required—just lengths of #22-30 AWG solid hookup wire. Molded-in holes let you mount QT units securely on any flat surface with 4-40 flat head screws, or 6-32 self-tapping screws, from behind panel.

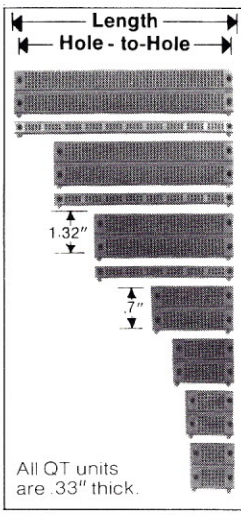
Economy—Sockets are priced as low as $3.00* Save more money by eliminating heat and mechanical damage to expensive parts, re-using components.

Speed—For fast circuit layouts, QT Sockets have 5 interconnecting tie-points per terminal: Bus Strips feature 2 separate rows of interconnecting terminals. Both connect and disconnect easily, without damage to socket or parts.

Visibility—All parts are instantly and easily visible and accessible, for quick signal tracing, circuit analysis and diagramming.

Durability—Higher-temperature sockets with abrasion-resistant, glass-filled plastic, rated better than 100°C. Screw-down-and-interlocked design provides high mechanical strength.

Reliability—Ruggedly designed to professional engineering standards, for heavy day-in, day-out use. Non-corrosive prestressed nickel-silver contacts insure more secure mechanical and electrical connections. Vinyl backing prevents shorting when mounted on conductive surfaces.



	Length	Hole-to-Hole	Terminals	Unit Price $*
QT-59S	6.5"	6.2"	118	12.50
QT-59B	6.5"	6.2"	20	2.50
QT-47S	5.3"	5.0"	94	10.00
QT-47B	5.3"	5.0"	16	2.25
QT-35S	4.1"	3.8"	70	8.50
QT-35B	4.1"	3.8"	12	2.00
QT-18S	2.4"	2.1"	36	4.75
QT-12S	1.8"	1.5"	24	3.75
QT-8S	1.4"	1.1"	16	3.25
QT-7S	1.3"	1.0"	14	3.00

See your CSC dealer or call
203-624-3103 (East Coast) or 415-421-8872 (West Coast)
major credit cards accepted.

*Manufacturer's suggested list • Prices and specifications subject to change without notice.

CONTINENTAL SPECIALTIES CORPORATION

44 Kendall Street, Box 1942
New Haven, CT 06509 • 203-624-3103 TWX: 710-465-1227
West Coast office: Box 7809, San Francisco, CA 94119 • 415-421-8872
TWX: 910-372-7992

interfacing a microcomputer to a pocket calculator

by **Michael Wimble**
6026 Underwood Ave. S.W.
Cedar Rapids, Ia. 52404

Originally, microprocessors were designed to emulate primitive minicomputer functions for service as terminal data handlers and industrial controllers. This is why microcomputers usually offer little in the way of built-in arithmetic functions. The recent emphasis on small computers offering high-level languages has created a demand for decimal arithmetic, floating point decimal, and trigonometric and statistical functions. Software implementation of these functions, although not necessarily complex, is usually tedious to develop, difficult to debug, and requires large amounts of expensive main memory.

calculator alternative

An alternative approach to software implementation of the desired functions is to interface the computer to a pocket calculator, thus providing a hardware extension at low cost. Before we consider the implementation of this technique, let's examine its deficiencies. There are two penalties. First, pocket calculators tend to be

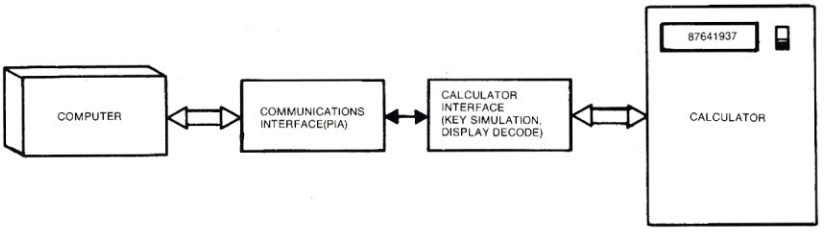


fig. 1. CPU to calculator interface consists of two separate interface subsystems.

table 1. Design goals for calculator interface.

1. The calculator is to remain useable as a pocket calculator. It must be designed to unplug from the interface for personal use.
2. Design requirement 1 and the need to develop a general interface which can be quickly attached to most popular calculators calls for an interface which will simulate keyboard entry.
3. The seven-segment calculator display is to be demultiplexed and decoded by the interface under computer control to provide addressable BCD digits for input to the computer.
4. All signal level-shifting required to connect the computer's communications interface with the calculator's keyboard and display are to be performed by the calculator interface.

slow; some calculators can barely perform ten multiplications per second. This speed limitation is usually acceptable for small systems which are not operating in a demanding real-time environment. The second objection is that we must design a special interface to connect the calculator to the computer.

choosing the calculator

Calculator selection should be made on the basis of a realistic assessment of the expected results. Extended computational delays may prove to be an intolerable nuisance factor, even in a personal computing system. First, list the functions you will need. Then measure the delay required by your intended calculator subsystem, being mindful of accumulated delays resulting from repeated use of a slow function. Naturally, selection of a sophisticated calculator offering greater precision (and therefore more digit enable lines) and numerous functions (each operated by a separate keyswitch line) will increase the cost of the interface.

table 3. Definition of Port No. 1 bits 4-7 sent from computer to the calculator.

PORT No. 1 bits 4 5 6 7	Definition of "enter digit" command	Definition of "enter operation" command	Digit addressed for "read display" command
0 0 0 0	0	(not used)	digit Ø(rightmost)
0 0 0 1	1	ADD	digit 1
0 0 1 0	2	MULTIPLY	digit 2
0 0 1 1	3	DIVIDE	digit 3
0 1 0 0	4	SUBTRACT	digit 4
0 1 0 1	5	PERCENT	digit 5
0 1 1 0	6	CLEAR/CLEAR ENTRY	digit 6
0 1 1 1	7	(not used)	digit 7 (leftmost)
1 0 0 0	8	EQUALS	(not used)
1 0 0 1	9	DECIMAL POINT	.
1 0 1 0	(not used)	(not used)	.
.	.	.	.
1 1 1 1	(not used)	(not used)	(not used)

interface functions

Fig. 1 illustrates the relationship of the two general interface functions required to establish communication between the calculator and the processor. A *communications interface* typically consists of one or more PIA's (Peripheral Interface Adapters), one-shots, address decoders and buffers. This interface handles data transferred to and from the processor bus. Therefore, its design is highly dependent upon the specific processor to be used. Since this type of interface is commonly offered as a standard package with most small computers, its design will not be covered here. The calculator interface, on the other hand, is a function of the calculator architecture and will be the subject of this article. Software routines for handling the data transfer at the computer side of the interface is also dependent upon the processor and the processor's communication interface. Therefore, software details will not be considered. It should be sufficient to point out that this software is much simpler than the arithmetic and trigonometric routines which we are replacing with the calculator and interface. Additionally, this software is primarily of the "LOAD and STORE" type and should be easy to write once the calculator is interfaced.

The following sections will discuss the design problems and hardware solutions will be offered. Information encoding and decoding, level shifting, and synchronization are covered. To aid in illustration of the techniques discussed, the specific example of interfacing to a simple five function Sears Roebuck calculator is presented. It is intended that the information presented will aid the reader when interfacing any of the popular

table 2. Typical bit level definitions for 8-bit input and 8-bit output ports used in a practical CPU communications interface connected to the calculator interface.

	PORT No. 1 — CPU TO CALCULATOR			PORT No. 2 — CALCULATOR TO CPU
BIT	USE		BIT	USE
0	Read display		0	Error indicator (user defined)
1	Enter command	command orders to calculator interface	1	Overflow indicator
2	Enter digit		2	Decimal point locator
3	(not used)		3	(not used)
4-7	For "read display" command mode, bits 5-7 specify digit to be displayed. For "enter command" mode, bits 4-7 specify the command (table 3), for "enter digit" mode, bits 4-7 specify the BCD digit to be entered (table 3)		4-7	BCD (binary coded decimal) representation of digit addressed by the "read dislay "command

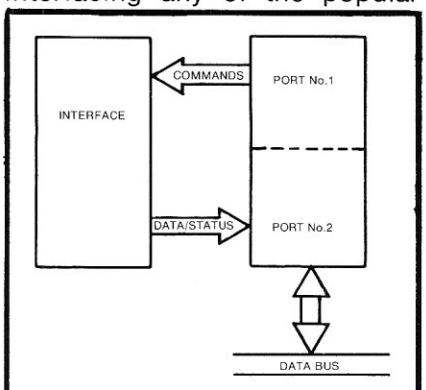


fig. 2. Typical two port communications interface employing a Peripheral Interface Adapter (PIA).

fig. 3. Port communications example showing a typical enter and read sequence. Computer commands the calculator to perform 3.4 X 7 and reads result on an 8 digit calculator.

PORT No. 1	PORT No. 2	
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	MEANING
0 1 0 0 0 1 1 0	*X X X X X X X X	enter CLEAR command
0 0 1 0 0 0 1 1	X X X X X X X X	enter digit "3"
0 1 0 0 1 0 0 1	X X X X X X X X	enter decimal point
0 0 1 0 0 1 0 0	X X X X X X X X	enter digit "4"
0 1 0 0 0 0 0 1	X X X X X X X X	enter MULTIPLY command
0 0 1 0 0 1 1 1	X X X X X X X X	enter digit "7"
0 1 0 0 1 0 0 0	X X X X X X X X	enter EQUALS command
1 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0	read digit Ø(rightmost)
1 0 0 0 0 0 0 1	0 0 1 0 0 0 1 1	read digit 1
1 0 0 0 0 0 1 0	0 0 0 0 0 0 1 0	read digit 2
1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	read digit 3
.	.	.
.	.	.
.	.	.
1 0 0 0 0 1 1 1	0 0 0 0 0 0 0 0	read digit 7 (leftmost)

*NOTE: X means bit value doesn't matter (not read)

pocket calculators with a general purpose microcomputer.

design goals

In order to insure that our interface will work with nearly every calculator on the market, our design goals (summarized in **table 1**) are to interface the calculator at the keyboard and display level. That is, our interface will appear to the calculator to be its keyboard, and the LED display control lines will connect directly to our interface.

I/O formats

Table 2 and **table 3** illustrate typical two port bit level definitions for the popular PIA type communications interface. The two port configuration is represented in **fig. 2.** Port No. 1 communicates information ("enter operation command," "enter digit," etc.) from the computer to the calculator. Port No. 2 communicates data and status information (BCD representation of the required digit, overflow indicator, etc.) from the calculator to the computer. **Fig. 3** shows a simple calculation transaction and the associated port data. The user should understand that it is the responsibility of the processor software to verify that valid data is sent to and received from the calculator. For this reason, bit zero of port No. 2 has been reserved for user definition for error checking (which depends upon the specific type of error and display information supplied by the chosen calculator).

calculator interfacing

Fig. 4 shows the functional overview of a calculator for interfacing purposes. The keyboard is simply an array of S.P.S.T. switches whose closure is sensed by the keyboard scanning circuitry in the calculator. Output from the calculator consists of two groups of data lines; one group specifies which of the seven *segments* are to be enabled in order to form a digit, the other group selects the digit *position* which is to display this information. For example, to display the number "34" the display *segments* for the number "4" are first enabled and then the rightmost digit is enabled. This display multiplexing results in each digit being turned on and off several hundred times per second, appearing as a constant display to the human eye, in addition to saving considerable expense via component reduction and extending battery life.

keyboard interface

Different calculator chips employ different methods for obtaining keyboard input. However, it is usually the case that the keyboard consists of S.P.S.T. switches. A general in-

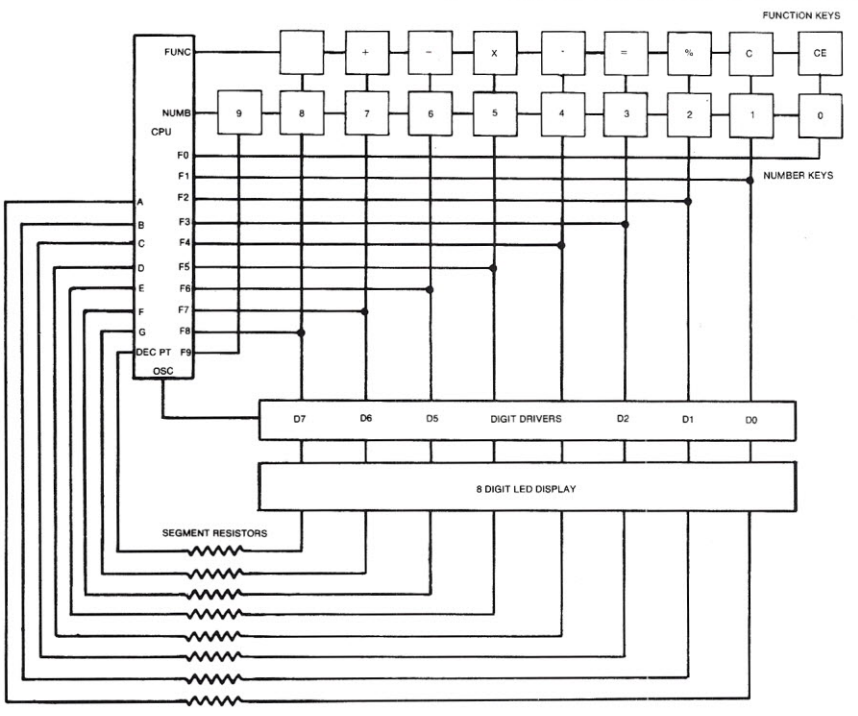


fig. 4. Functional overview of a pocket calculator.

table 4. Relationship between keys pressed and keyboard connections shorted as determined with an ohmmeter.

KEY	CONNECTIONS SHORTED BY THIS SWITCH
Ø	f & l
1	a & l
2	b & l
3	c & l
4	d & l
5	g & l
6	j & l
7	e & l
8	h & l
9	i & l
.	i & m
+	a & m
×	b & m
÷	c & m
−	d & m
C/CE	f & m
=	h & m
%	e & m

terface can be implemented with only this knowledge by simulating keyswitch closure in parallel with actual keyboard keyswitches.

The first step is to determine the exact keyboard format used by the chosen calculator. One of three formats is commonly used:

1. Each switch is independent of all others, requiring two lines to be connected to each switch from the scanning chip or circuit.
2. The switches are diode encoded. There will be one *common line* for all switches and one separate line for each switch.
3. The keyswitches are scanned in multiplex fashion. This is the *most common* method and is illustrated in **fig. 4.**

If common lines *are* used in the keyboard configuration our interface circuitry will be simplied.

determining keyboard format

Like many inexpensive calculators, the calculator chosen for this article had a connector across the top of the printed circuit board which forms the contact panel for the keyboard. The cable connecting to the keyboard is shown as the righthand bundle of wires in **photo 3.** By labeling the keyboard connector pins "a" through "m" from left to right and using an ohmmeter, the relationships between keys pressed and connections shorted by these keys were determined as shown in **table 4.** As shown, there are two common lines employed (similar to the calculator shown in **fig. 4**). Line "l" is common to all *digit* keys; line "m" is common to all *function* keys. Clearly, this type of keyboard is equivalent to two single pole, multiple throw switches. If this common line configuration is not present in your chosen calculator, it may be best to view the keyboard simply as a collection of independent S.P.S.T. switches.

device selection

There are many methods we could use to simulate the keyboard. Relays could be connected in parallel with the switches but the cost and board space required would make this a poor choice. In addition, transistor drivers may be required to activate the relays. Transistor switches could be used to simulate key closures.

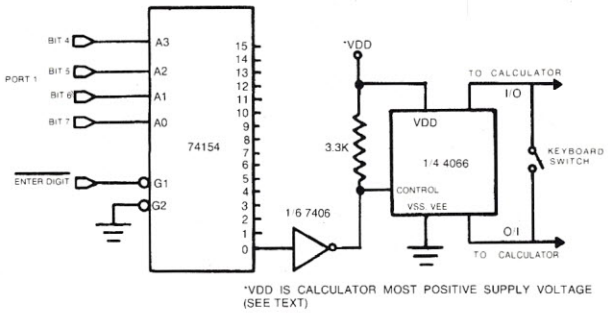


fig. 5. Section of a keyboard interface showing one method for CPU to keyswitch interfacing to be used with keyboards not using a common keyswitch line.

However, there is a better solution which is less expensive and more compact. The practical and least expensive method is to use the new CMOS analog switches. Several hobby supply companies are currently offering these for less than one dollar in quad (four units in one) packages.

level translation

Before we examine the CMOS analog switch as a simulation device for keyswitch closure, we must consider the problem of voltage level translation, specifically the translation required between the calculator and the CPU. The CPU generally communicates at TTL signal levels, while the calculator may use one or more MOS voltage levels. If required, contact the *calculator chip* manufacturer to obtain pinouts and logic levels.

Do not feel that the manufacturer's recommendations must be blindly followed. In particular, shift the *relative* logic levels to a more suitable range if desired. If, for example, the calculator uses a +5V, ground, and −12V supply system, it may be more convenient to translate these levels to +17V, +12V, and ground. Notice that we haven't changed the *relative* logic levels with this translation. Establishing a frame of reference where all logic levels are positive with respect to ground simplifies the design of the interface.

transmission gates

The CMOS analog switch

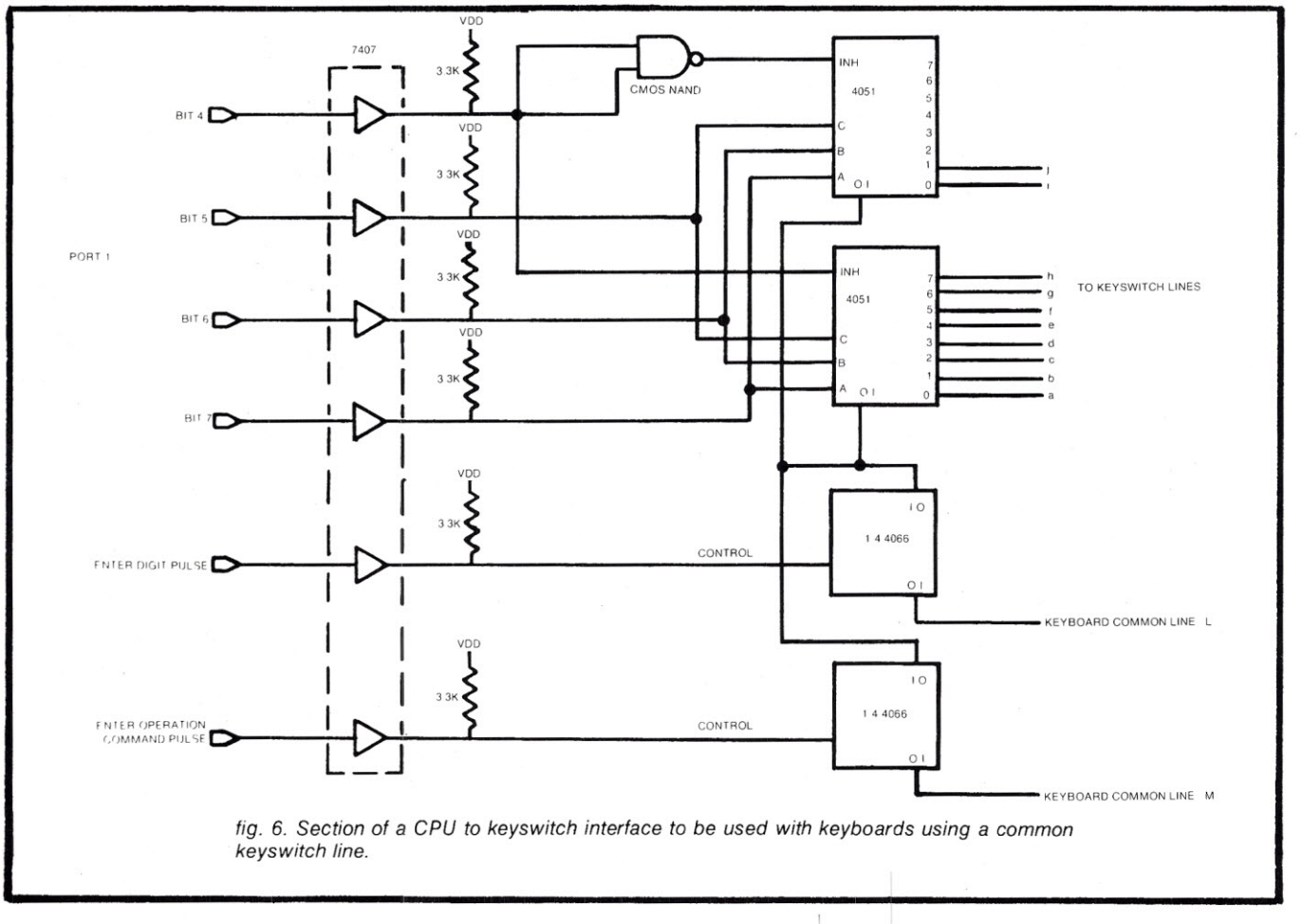


fig. 6. Section of a CPU to keyswitch interface to be used with keyboards using a common keyswitch line.

mentioned previously employs *transmission gate technology.* The gate is a solid state equivalent of a relay. When the control line is high, the gate will pass analog signals *bidirectionally* with very little resistance, typically from 80 to 150 ohms. When the control line is low the gate acts as an open circuit (10^{12} ohms). The gate will, however, pass only those signals whose amplitude does not exceed its supply voltage. For example, if the gate is powered by a +5V and a −5V supply, then signals exceeding this range will be badly distorted. This is why it is wise to identify the most negative and the most positive supply voltages on the calculator chip and then assign the most negative level to be ground. All chip voltages will then be positive with respect to ground. If we connect the transmission gate to take its power between the level we have defined as ground and the most positive voltage on the chip, then we are assured that any possible voltage level present on the keyboard will be passed by the transmission gate and it will be able to simulate a keyboard switch.

The transmission gate power supply voltage levels determine what the control voltage must be in order to produce the desired "on" or "off" condition. In this article the control voltage for the "off" condition must be closed to ground, and the "on" condition requires that the control voltage be close to the positive supply voltage. By using a 7406 or a 7407 open collector gate, this level translation is easily achieved.

keyboard simulation

If the keyboard is organized as a collection of S.P.S.T. switches, CMOS devices such as the 4066 or the 4016 (similar to the 4066, but has a higher "on" resistance and is less expensive) can be wired in parallel with the keyswitches. A partial interface employing the 4066 in parallel with a keyswitch which does not use common lines is illustrated in **fig. 5.** As shown, the 74154 4-line-to-16-line demultiplexer is used to *decode* the BCD digit lines from port No. 1 (review tables 2 and 3 if necessary). The "enter digit" pulse is used to enable the digit demultiplexer, causing the appropriate digit to be entered into the calculator. The 7406 converts the low going pulse to a high going pulse (since it's an inverter), and performs the desired level translation for the transmission gate. Since, as previously discussed, the transmission gate

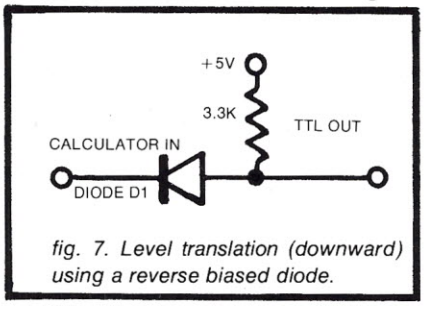


fig. 7. Level translation (downward) using a reverse biased diode.

takes its supply voltage between the most positive and the most negative chip voltages, the pull-up resistor on the open collector gate is used to raise the logic "1" voltage for the control gate past the threshold voltage.

common line circuit

Fig. 6 shows an interface for common line keyboards. The 4051 8-channel multiplexers select one of the non-common ("a" through "k") keyboard lines while the correct common line is selected via a 4066 transmission gate enabled from either the "enter digit" or the "enter operation command" line from port No. 1.

One final note about the keyboard interfacing circuitry; the "enter digit" and "enter

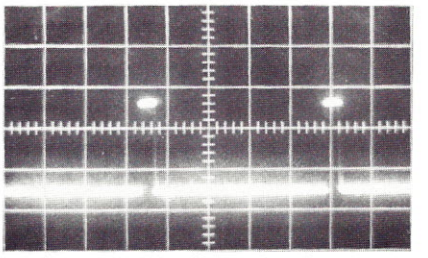


photo. 1. Waveform showing the digit and segment logic voltage levels for the calculator chip. Y=2V/cm, X=0.5ms/cm. Baseline is the bottom border of the grid.

operation command" lines must be pulsed to enter the information into the calculator. This is usually performed by a one-shot or similar circuitry in the interface. It may also be timed with a software loop. Some calculators may require that the enable line be low for some period of time before the next digit or command can be entered.

display interfacing

As nearly all the popular calculator chips produce the same type of output, interfacing to the display is more standard than interfacing the keyboard. Calculator results will be expressed as a multiplexed seven-segment code. Our interface must accomplish two goals: convert the multiplexed seven-segment information into addressable BCD information, and translate the calculator display control voltages to TTL levels used by the CPU's communications interface.

The downward voltage trans-

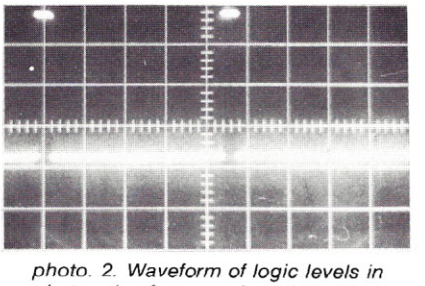


photo. 2. Waveform of logic levels in photo. 1 after passing through a CMOS NAND gate. Note that the baseline is the third grid line from the bottom.

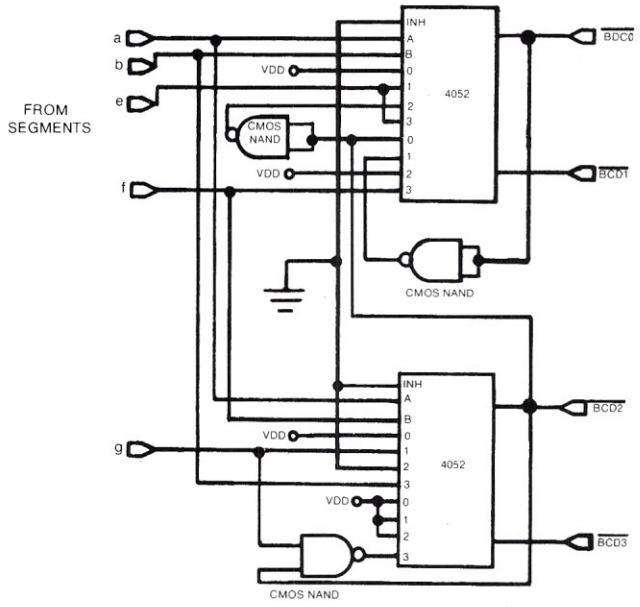


fig. 8. Simple circuit for 7-segment to BCD converter. 4052's are differential 4-channel multiplexers; two control lines select one of four pairs of inputs.

lation required at the display interface is at least as simple as the upward voltage translation required for keyboard interfacing. **Photo. 1** shows the digit and segment enable logic levels as they exist at the calculator chip. A logic "1" is about 7.2 volts while a logic "0" is about 3 volts. A simple resistive voltage divider could be used to translate these levels to the required TTL levels. However, this would also reduce the voltage margin *between* the logic levels, thus lowering the noise immunity.

Op amps could be used to provide a voltage offset to translate the logic levels while retaining the noise margin, but this choice would be expensive overkill and would require many discrete components. The simplest method is to use a reverse bias diode as shown in **fig. 7.** When the logic level illustrated in **photo. 1** is passed through this circuit, the diode clips the 7.2V level to the required 5V TTL level. The 3V logic level remains at 3V since the diode does not clip at this low level. A TTL device would, of course, recognize this 3V level to be a logic "1" which it is *not.* In order to translate the input logic "0" level to the correct TTL level, we could simply pass the signal through a CMOS NAND gate. The result is shown in **photo. 2;** the logic "1" level is still greater than 7V, but the logic "0" is now 0.1V and will be recognized as a valid logic "0" by any TTL device. The 7V logic "1" is then passed through the diode circuit shown in **fig. 7** to translate this signal to a valid TTL logic "1".

7-segment-to-BCD

Rather than translate logic levels for seven 7-segment enable lines and eight (or more) digit position enable lines, it is more economical to perform the 7-segment-to-BCD code conversion first, and then to perform the logic level translation on the four BCD lines. There are several methods which may be used to convert 7-segment code to the BCD format but only one is recommended here. NAND/NOR conversion is simple, but requires many IC chips. ROM lookup is ideal for those readers who have access to ROM programming facilities, but will not lend itself to a good solution for many hobbyists.

Fig. 8 shows a fairly simple and commonly used circuit for 7-segment-to-BCD conversion. The output of this circuit is passed through NAND gates to yield the previously mentioned logic level translation. The only remaining requirement for completing the calculator interface is to add the digit addressing circuitry.

Fig. 9 represents a complete display interface. The 4051 8-channel demultiplexer selects one of the digit enable lines to synchronize the conversion to the BCD format. Bits 5, 6, and 7 of port No. 1 give the digit address (as specified by the computer) and the "read display" line is raised. When the calculator display multiplexing circuitry pulses the appropriate digit enable line, the BCD data

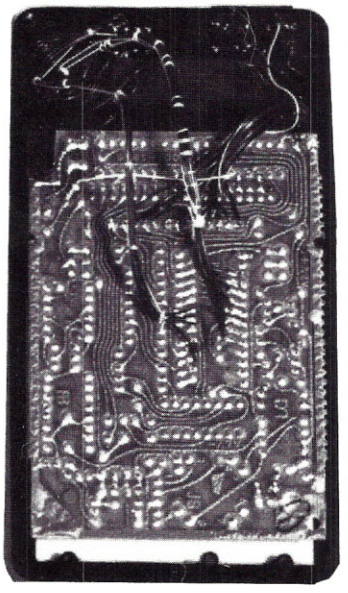
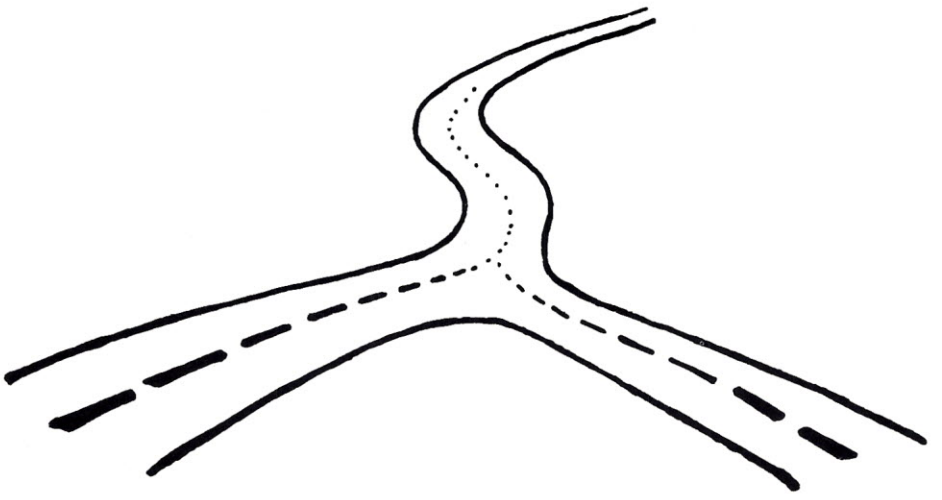


photo. 3. Shown in this photograph are the three cables used to interface to the keyboard and display.



MERGING MAGAZINES

If you are a subscriber to MICROTREK you will automatically begin to receive PERSONAL COMPUTING which will contain the articles and editorial material formerly featured in MICROTREK. MICROTREK is, as discussed on page 6 of this issue, no longer being published as a separate magazine. We believe that this merger will greatly benefit MICROTREK subscribers. When you receive PERSONAL COMPUTING, please send comments and specific suggestions for improvements to the MICROTREK Editor. We will strive to include and present the type of articles you would like to see. Authors, take note that our need for good articles is now greater than ever!

MERGER DETAILS

As a MICROTREK subscriber, you may fit into one of three subscription situations:
CASE 1: You subscribed for 12 issues of MICROTREK and have received 2, including this issue. You will receive 7 issues of the bimonthly PERSONAL COMPUTING (total value based on regular subscription rates to both magazines is $10.99).

CASE 2: You subscribed to 14 issues of MICROTREK (offered on a limited basis at conventions) and have received 2, including this issue. You will receive 8 issues of the bimonthly PERSONAL COMPUTING (total value based on regular subscription rates to both magazines is $12.32).

CASE 3: You have subscribed to both MICROTREK and to PERSONAL COMPUTING. In this case your subscription to PERSONAL COMPUTING will be extended to cover the merger issues plus your regular subscription.

PLEASE NOTE: If you have subscribed to both MICROTREK and PERSONAL COMPUTING, fill in your subscription numbers and name and address on the special card bound into this issue next to the inside back cover. Mail this card as SOON AS POSSIBLE to avoid any duplication in the issues sent to you. Use this card for any questions or comments you may have regarding this merger.

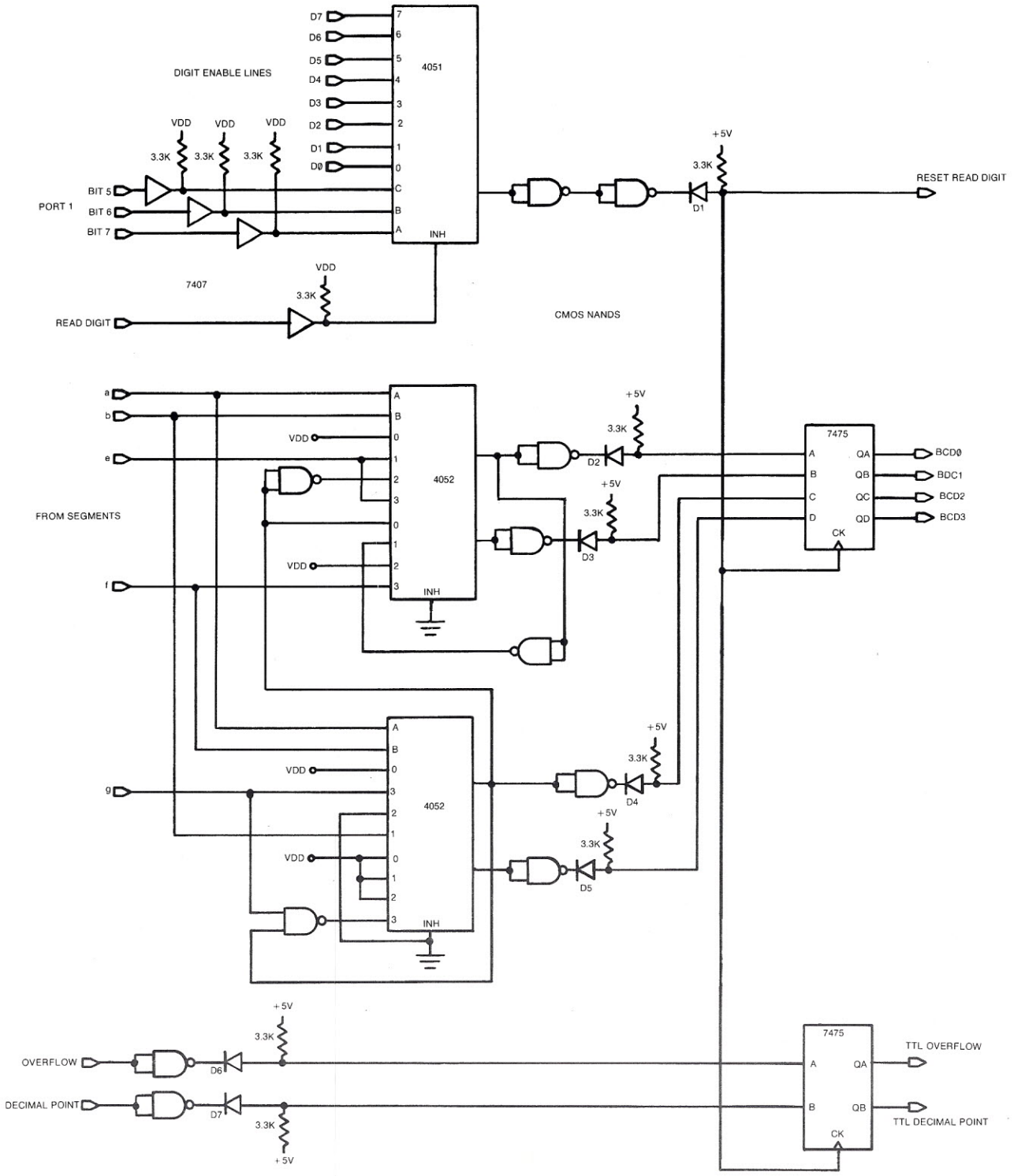
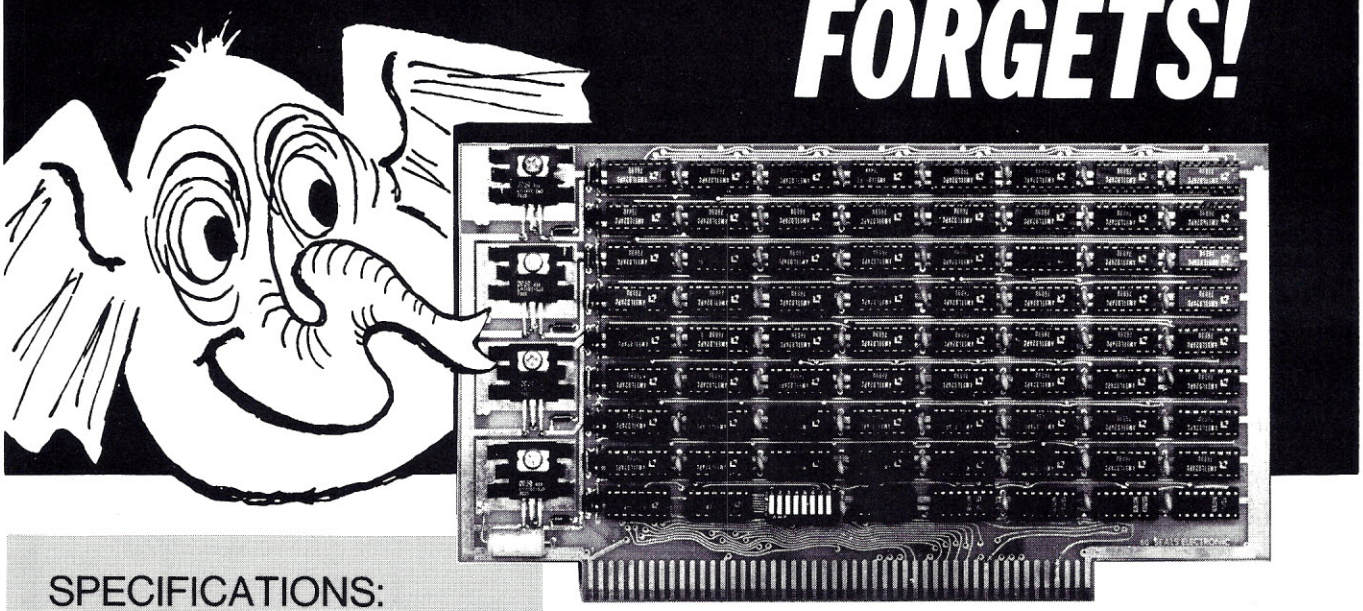


fig. 9. Complete calculator display interface with 7-segment to BCD converter, digit addressing (selects digit to be input to computer), and level translation.

The first 8-K that NEVER* FORGETS!



SPECIFICATIONS:

8K SC - 8 Specifications:

Access Time:	500 ns Max. (225 max on request)
Current Req:	Less than 200 ma per 1024 words maximum
Memory Chip:	AMD 91L02 APC **(low** power 1K x 1)
Voltage Supply:	+5 to +10 volts
Battery Standby:	1.5 to 2 Volt, Automatic power loss sensing circuit. Eliminates need for switches.
Address Select:	8 ea. Spst. switches in a Dip IC package. (No longer any need for a soldering iron to change address.)
+5 Volt regulated:	4 ea. 7805 regulators with individual heat sinks to run cooler.
Wait States:	NONE! Your wait light will not burn because of a memory wait state.

ALL ADDRESS, CONTROL, AND DATA OUT LINES FULLY BUFFERED

Circuit Board

Double sided, G10 glass epoxy board
Plated through holes. 5 mil. tin minimum
Solder reflow processed
Solder mask on both sides of PC board
Component lay out silk screened on component side of PC board
Gold plated edge contacts
No jumper wires used
Professional layout techniques used

ALL ADDRESS, CONTROL, AND DATA OUT LINES FULLY BUFFERED

•

QUANTITY, DEALER, AND CLUB INQUIRIES INVITED

•

PLUG IN-COMPATIBLE WITH ALTAIR® AND IMSAI®
IC SOCKETS INCLUDED

$295.00 Kit — $394.00 Assembled — $2.00 Shipping and handling

•

*LIMITED TO CAPACITY OF STANDBY BATTERY

SEALS ELECTRONICS®

BOX 11651, KNOXVILLE, TN. 37919

Enclosed is $_____ ☐ FULL AMOUNT ☐ 25%; BALANCE C.O.D.

☐ BANK AMERICARD/MASTER CHARGE #_____

PLEASE SHIP THE FOLLOWING:

☐ 8KSC KITS .....@ **$295.00** each

☐ 8KSC ASSEMBLED .....@ **$394.00** each

TO_____

STREET_____

CITY_____ STATE_____ ZIP_____

ADD $2.00 SHIPPING AND HANDLING


Termina Computer
Termina Computer
Processor Technology

The first complete small computer.

As you thumb through this magazine, you'll see a lot of ads for small computers. For $600 you can find a pretty good box with a power supply, four slot mother board, CPU module, and all the expected lights and switches.

But you know what? It won't work.

That's because in order to make the computer go you have to buy memory—normally both read-write (RAM) and read-only (PROM), interfaces to the outside world (parallel, serial, and cassette), keyboard, video display module, and software.

Add this all up and it's going to cost you at least $1,400 for a complete system.

Got the picture?

Now listen to this. The remarkable new Sol-20 Terminal Computer will give you all of the above... plus more!... as *standard equipment* for just $995, in kit form. This is because the Sol-20—like no other small computer—was designed from the ground up to be complete.

Here's what the Sol-20 includes as standard equipment:

- 8080 microprocessor.
- Video display circuitry.
- 1024 words of static, low-power RAM.
- 512 words of preprogrammed PROM.
- Custom 85-key solid-state keyboard.
- Audio cassette tape interface.
- Both parallel and serial interfaces and connectors.
- Power supply.
- A beautiful case with solid walnut sides.
- Software that includes a preprogrammed PROM personality module and a cassette with BASIC-5 language plus two sophisticated computer video games.
- Full expansion capability with all S-100 bus (Altair/IMSAI/PTC bus) products.

It's a handsome terminal or computer that will even look good in your living room or office. Small systems start at just $475.

Full expansion capability

The Sol-20 system can be tailored to your applications using the complete line of peripheral products from Processor Technology. These include the video monitor, audio cassette and digital tape systems, dual floppy disc system, memories, and interfaces... plus all other peripherals compatible with the S-100 bus.

The Sol-20 greatly simplifies the computer-buying process. It's a splendid package that will excite both present and prospective computer owners.

Our brochure tells all. Write for it today.

Processor Technology, Box P,
6200 Hollis Street, Emeryville, CA 94608.
(415) 652-8080.

The Sol-20 expandable system.

Processor Technology Corporation

photo. 4. 16 pin DIP sockets mounted on the calculator case.

generated by the 7-segment-to-BCD converter is latched into the 7475 quad latch and a pulse is generated to reset the "read display" command line, assuring us that the digit is converted only once. This reset pulse may be sent back to the computer via an unused PIA line or it can be used with a one-shot to activate a hardware reset. The choice depends upon the particular communications interface used with the computer.

error flags

Other enhancements shown in **fig. 9** include capturing the decimal point and the overflow indicator. The decimal point indicator may be read directly from the chip. The overflow indicator may have to be decoded from the display for some types of calculator chips.

interface mechanics

Having discussed interfacing methods for both calculator input and output, there remains the details of the physical implementation. First there is the matter of interface connectors. Our goal of being able to use the calculator as it was originally intended requires an interface connector which may simply be unplugged from the calculator.

Photo. 3 shows the three miniature cables used to connect the calculator to the interface sockets. The larger cable at the upper right contains the keyboard lines and the power supply leads. The center cable consists of segment enable lines, while the left cable contains the digit position enable lines.

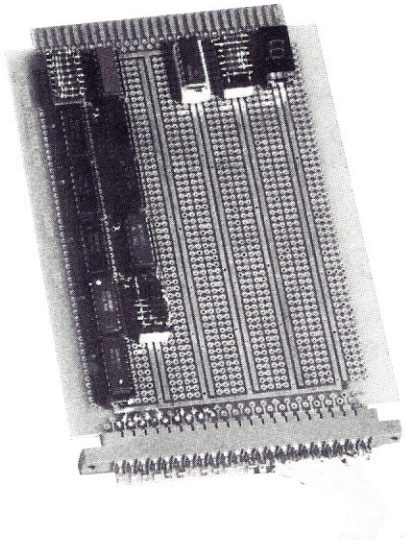


photo. 5. Author's prototype interface with built-in digit display for debugging.

As shown in **photo. 4,** DIP sockets were mounted on the top of the calculator case which was drilled to pass the cables. These 16 pin sockets were simply glued to the case. With this method very little space will be consumed by the interface connections.

power considerations

The calculator battery power line was routed through the interface sockets so that a shorting plug can be used to disconnect the calculator supply, eliminating any problems with power supply conflicts. When attached to the interface, the calculator is powered by the computer or interface supply and when the shorting plug is inserted, the calculator is powered by its internal batteries.

Do not assume that a 6V or a 9V battery pack means that the calculator actually runs at 6V or 9V. The sample calculator uses a 4V NiCad battery pack to supply a 7.8V chip voltage via the DC-to-DC converter in the calculator.

prototype interface

The interface prototype constructed by the author is shown in **photo. 5.** This interface contains a seven-segment decoder, display, and current limiting resistors on the upper right-hand portion of the board to display the contents of the 7475 quad latch. This makes it easy to check for correct digit addressing and to verify that the 7-segment-to-BCD conversion was performed as expected. This built-in debugger is especially helpful when developing the software to operate the interface.

Unless you have access to information detailing the logic levels and operation of the calculator you have chosen to use, I suggest that you prototype your interface with wire wrap sockets; it's hard to determine for certain if your interface gate is going to work with the calculator without exceeding its drive capabilities until it is actually tried. In addition, long connecting cables may cause ringing problems. The pocket calculator interface should prove to be an interesting and useful addition to your computer.

buyer's report:

the Digital Group System

by Steven R. Woodall
34 Fendale St.
Franklin Square, N.Y. 11010

"It plays the *Star Spangled Banner* on the radio while it draws a picture of the U.S. Flag on the TV," I argued. "We can't afford another expensive gizmo," countered my wife sharply. "And besides, it's a lot cheaper to watch the Bicentennial Minutes," she added. We were, of course, discussing the merits of my proposed purchase of a Digital Group (DG) System computer. Being a confirmed computerholic, I hocked my camera and bought the full-blown DG Z-80 system despite domestic objections.

Anything that pays for itself makes sense to me. That's why I chose the Z-80 version of the DG system (they can sell you just about any CPU board you might wish for). The additional cost of

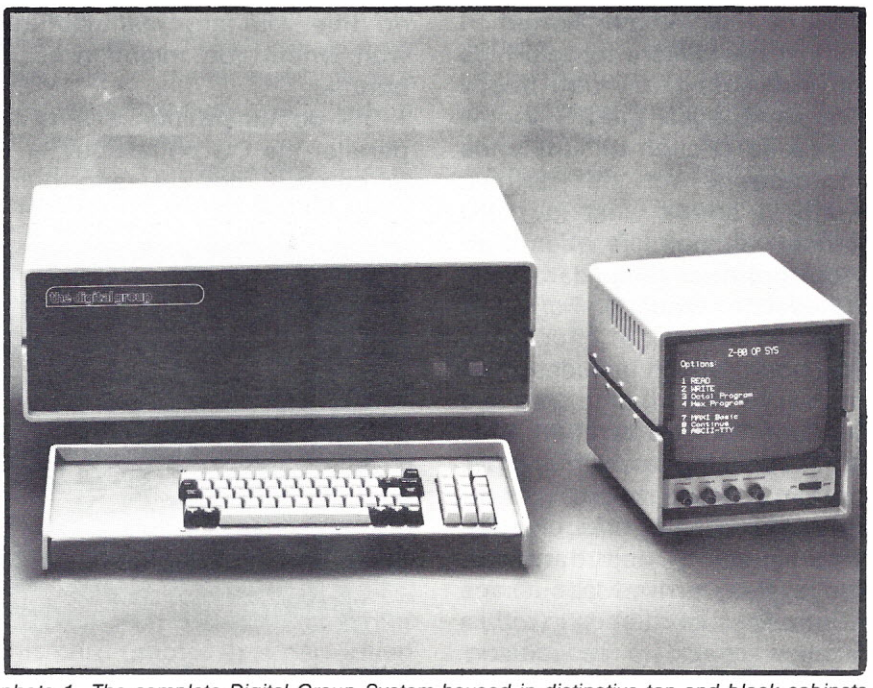


photo 1. The complete Digital Group System housed in distinctive tan and black cabinets. Promised shipping date to customers is early February.

powers of 2 which will add up to

approach to card-level design.

A 2N5129 video combiner pro-

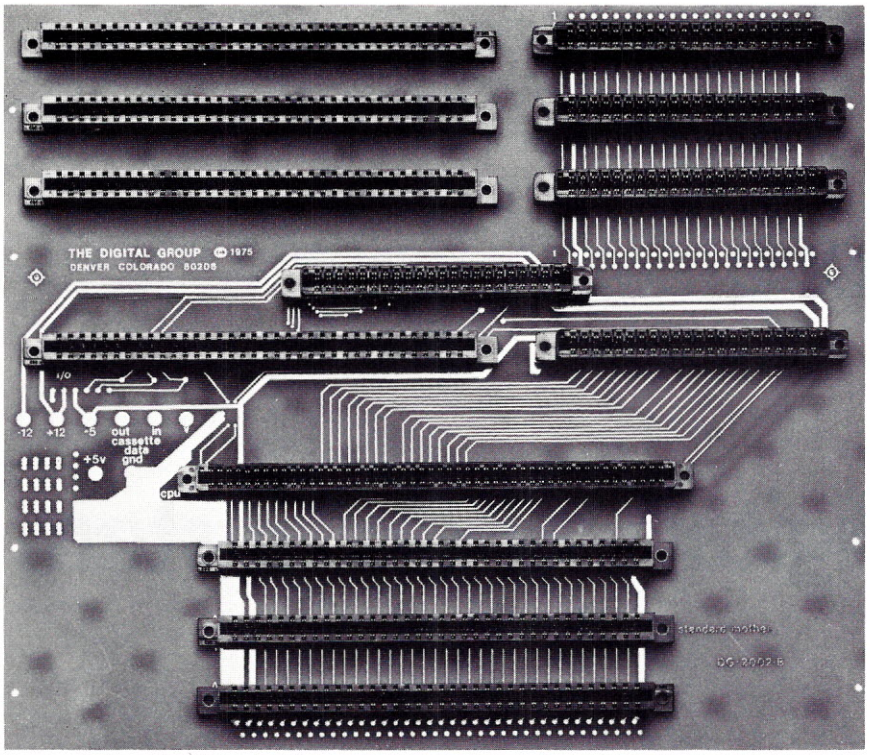


photo 2. This is the Standard Motherboard. It supports 26K of memory, 16 input ports and 16 output ports, in addition to the CPU, TV/Cassette cards.

the Z-80 CPU version is offset by the reduction in memory used by this super powerful chip. So there, 8080 fans — may the Bird of Paradise eat your 8228 system controller.

Since, as I mentioned, I hocked my camera to help hide the true cost of my new hobby from my spouse, the DG people were kind enough to supply the photographs for this article. **Photo. 1** shows what my CPU cabinet will look like when it arrives (promised ship date from DG is late January). For $895 (shipping included) I bought the Z-80 board (which includes 2K of memory), an I/O card, TV Readout/Cassette card, 8K RAM card, 12 Amp power supply, Standard Motherboard, Standard CPU cabinet, fan, binder, documentation, and hardware. The system arrived about one month after I ordered it (with a certified check to reduce delays). If you have a pressing need (or can't stand suspense), call the DG (303-777-7133) and get the current estimated shipping dates. I have found them to be exceptionally forthright and courteous. Call and get acquainted — and I suggest you do this with any manufacturer with whom you might invest a bundle.

DG documentation seems to parallel the DG philosophy — it is concise, correct, and complete. It's not a dissertation on microcomputer theory, but it tells you what you need to know to put the system together and get it running. Each board is documented in a separate section which is well written and very carefully organized. My only objection is that the few component value corrections which show up in the documentation might have been noted in the very first part of the documentation. Detailed board debugging is included.

Since the motherboard embodies the character of the entire system, we will evaluate this board first. Incidentally, the order in which we will examine the boards is the assembly order recommended by the DG.

the Standard Mother

Photo. 2 shows the 'Standard Motherboard' fully populated with the appropriate edge connector sockets. Since each board kit comes with the required socket to mate it to the motherboard, the Standard Motherboard is supplied bare. Molex connectors are recommended for slipping over the wire wrap sockets supplied with each I/O kit to connect the input and output ports to peripherals. These are *not* included with the kits since the number of connec-

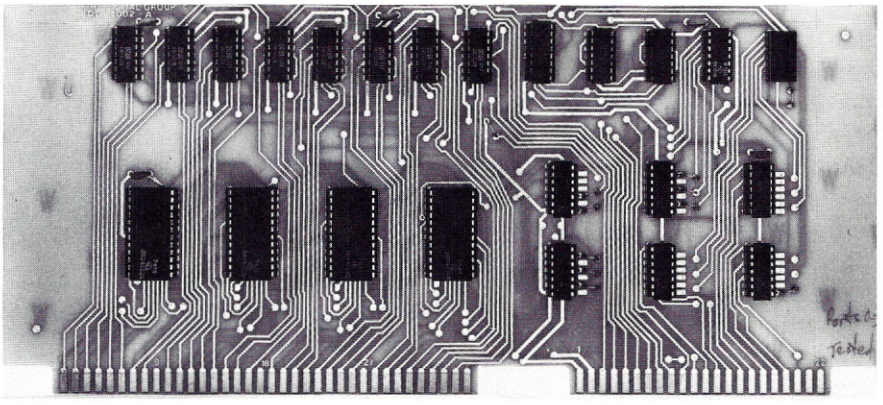


photo 3. Four latched 8-bit output ports and four gated (open collector to bus) 8-bit input ports reside on this board. Address jumpering permits addressing these ports at any four consecutive locations in the 65K range.

tors required depends on the number of I/O ports you wish to establish. For $35 you can buy a Molex connector kit, complete with flat cable, from the DG. Add $8.95 if you need the crimp tool. Refer to the DG Flyer No. 5 for details regarding the Molex connectors.

I/O Card features

The DG I/O card is not a fancy programmable type. Instead of inscrutable Peripheral Interface Adapters (the title is bigger than the chip), we find a simple, respected workhorse — the 74100 TTL latch — serving as the output port element, and the even simpler 7404 open-collector dual-input NAND functioning as the input port element. That's what I like about the DG design; it's so simple that you will understand it even when it doesn't work. If you've ever had to trouble-shoot some of the complex, kludgey systems designed by people who worship large-scale integration circuitry, you will be soothed by the shocking simplicity of the DG system.

Address lines terminate in 16 7404 gates which are in turn connected to 16 7404 gates. This series buffer configuration permits the user to pick off the inverted or non-inverted address line signal for the desired decoding. The four 8-bit input and output ports on the I/O card can be set up only as a block of four sequential (contiguous for you picky folks) locations; 0-3, 4-7, etc. Here's the procedure: determine the number of the highest port to be on the card. Then find the combination of the powers of 2 which will add up to this number. For example, if ports 24, 25, 26, 27 are the ones you want, figure that 27 is a binary 11011. Therefore, tie the address straps No. 0, 1, 3, 4 to logic one pins (next to the inverter chips), and strap No. 2, 5, .., 15 to logic zero pins. All four ports are now properly decoded. And if this isn't simple enough for you, the DG documentation has a map of the straps with an example to make sure you understand how easy it is to set up the addressing.

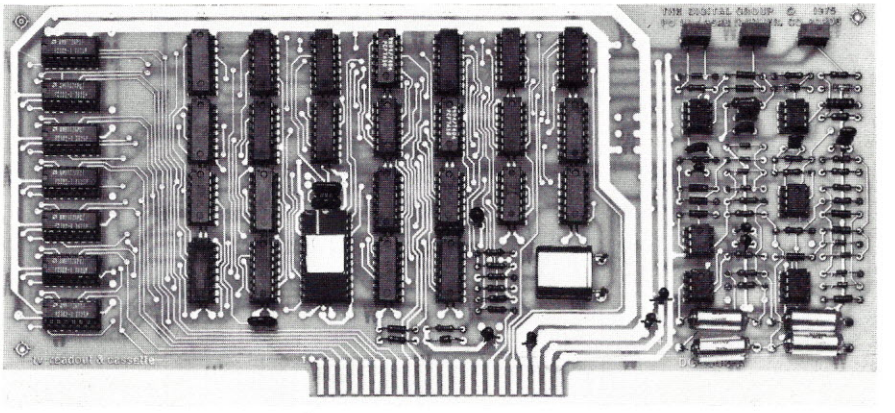


photo 4. Here we have the TV Readout/Cassette card. The readout employs a 7 X 13 dot matrix to produce each character for enhanced resolution. The display buffer consists of the RAM at the left, while the cassette interface active filter is on the right.

Photo. 3 reveals the clean, open layout which should make it a snap for you to construct. I spent only a few hours on this board and it worked right off. Just remember to mark the card so you won't put it into the wrong slot — each I/O card is *of course* given a unique block of addresses, and the external connections to I/O devices are wired to these *specific* I/O slots.

TV Readout/Cassette card

Since all address decoding for this card **(photo. 4)** is performed on the I/O card, the DG decided to combine both the TV readout and the cassette interfaces on this single card. We will examine the TV section first.

Getting to the point with minimum frills sums up the DGs' approach to card-level design. The simple but effective circuitry of the TV Readout is a good illustration of the reliable, low-cost character display which results from the implementation of this philosophy. Exotic functions such as reverse lettering (black letters on a white background) are not included, but a full range of upper and lower case ASCII characters, math symbols, Greek letters, and special symbols make this a very capable display for general programming and text editing (DG offers a Text Editor Cassette for $7.50).

An unusual feature in this TV typewriter is the hardware Write position counter. This controls the location of the next character to be written on the screen. Casting this function in hardware permits the use of this display with any ASCII encoded keyboard in a stand-alone mode, freeing the processor from the chore of character writing to the display buffer. If, on the other hand, software control of the cursor is desired, character positioning becomes cumbersome because the Write counter must be stepped until the desired location is arrived at. Incorperation of a circuit to preset the Write counter directly from the processor would overcome this objection nicely.

A 2N5129 video combiner provides a composite video signal (3V p-p) which may be coupled into a commercial monitor, a TV set modified to serve as a monitor or a simple modulator/oscillator circuit for use with an

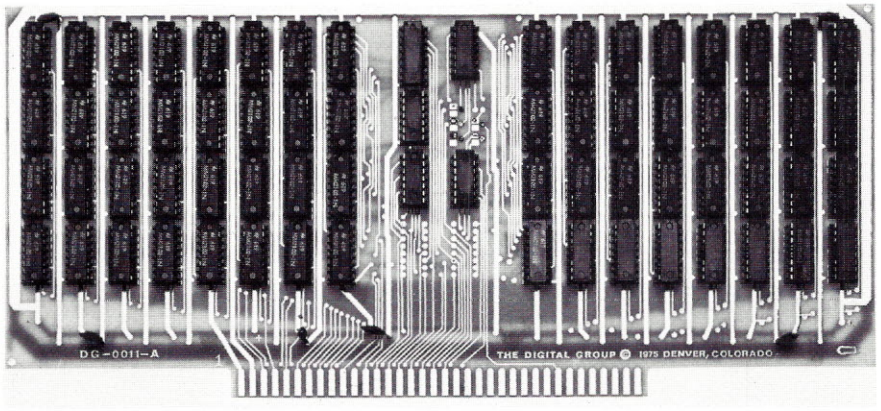


photo 5. The 8K RAM board shown here takes about four hours to assemble. A considerable number of uncommitted connector pins are provided, making it possible to support a 16 bit processor if desired.

unmodified TV set. The latter is possible because the display signal has a bandwidth of slightly more than 6 MHz.

Construction of the TV Readout portion of this card requires considerable care due to the high density of the data paths. It took me about 5 hours to assemble this part of the board. Unfortunately, in my haste to see something on the screen I overlooked the soldering of an IC pin. When the system was first turned on, writing to the display buffer produced no change on the screen. I was lucky — impatience cost me only an hour of Sherlock Ohmsing — you may not be so fortunate. One last piece of advice; epoxy the crystal to the card to ruggedize your system.

Cassette interface

1100 bits per second is a fairly high data transfer rate for an inexpensive mass storage system. This is the primary feature of this cassette interface. Data is recorded by converting a logic one (called a Mark) to a 2125 Hz. tone and a logic zero (Space) to a 2975 Hz. tone. This is accomplished with an extraordinarily simple 566 voltage controlled oscillator (VCO) circuit. The frequencies chosen for use with this frequency shift recording technique are widely spaced and are unrelated harmonically, providing good signal/noise ratio recording with any reasonably stable cassette recorder. Bit 0 of port No. 1 is used to input the data to be recorded. Memorex MRX2 or Maxell UD tape is recommended for optimum reliability. In addition, the DG recommends not using the first minute or so of any commercially prepared tape as their experience indicates a statistically high drop-out rate in this area of the tape due to handling.

data recovery

Audio tone-to-digital data conversion is performed by several op amp stages. The raw recorded output is clipped, passed by one of two active bandpass filters to a full wave detector, low-pass filtered to remove any remaining audio component, and adjusted to the proper TTL level for input to the processor.

The open layout of the cassette interface portion of this card reduces the risk of solder shorts. The lack of silk screened component labels on the card, however, makes it slow going (component designations are specified in the assembly documentation).

Alignment and construction of the cassette section required about three hours to complete. Accurate alignment is required for optimum reliability, especially at a rate as high as 1100 bits/second.

The VCO circuit requires two simple adjustments; set the Mark potentiometer to 2125 Hz. with a logic one applied to the interface input; then apply a logic zero and adjust the Space potentiometer for 2975 Hz. at the output. A final tune-up is performed by programming the computer to supply the logic levels to the interface input and again trimming the Mark and Space potentiometers. The cassette interface is now ready to record data.

Alignment of the data recovery section of the cassette interface requires three simple adjustments. A scope and a source of clean audio signals (2125 Hz. and 2975 Hz.) will be needed. The two active filters are peaked at their desired bandpass frequencies by paralleling trimming resistors on the filters. The final adjustment involves setting the Read Offset potentiometer for a Mark Hold condition when no tone is applied to the interface input. The Cassette interface is now ready for use. If you wish, the DG will make the final alignment for $20.00. If a 'fix-it' is also required (sorry — basket cases will not be accepted), send $30 when returning the card.

8K Memory Card

If you've seen one 8K RAM card, you've seen them all. While I am quite impressed with the practical design approach which the DG has employed throughout their products, I was a little disappointed with their ultra-

plain 8K Memory card. There is no convenient DIP switch to select the address decoding (the board can be *strapped* to reside in one of 8 blocks of memory; 0-8K, 8K-16K, etc.). My main objection, however, is that there is no voltage regulator on this board. One voltage surge on the power line and it's 'Goodbye Mr. Chips.' I wish someone would pass a law to make it mandatory to put voltage regulators on memory boards and CPU cards. 2102-1's are standard with this card. That means that each 8K board will draw nearly 2 Amps.

Since there are 64 1K X 1 memory chips, they are wired up to produce an 8K X 8 matrix. The address lines are buffered with 7404's. There is a section of a 7420 four input NAND gate used to degate the RAM, permitting use of valuable lower address space by both RAM and ROM.

I would recommend that you construct this board in two sittings; do one side of the board, move onto another board, then return and finish the 8K RAM board. This will help you avoid fatigue caused by the never-ending task of soldering row-upon-row of DIP pins. Without this break it is very easy to overlook a pin to be soldered or worse still, a fatal bridge. If you do make an error on this board, the DG group has concocted a supper nifty memory test program to pinpoint the trouble, as we shall see.

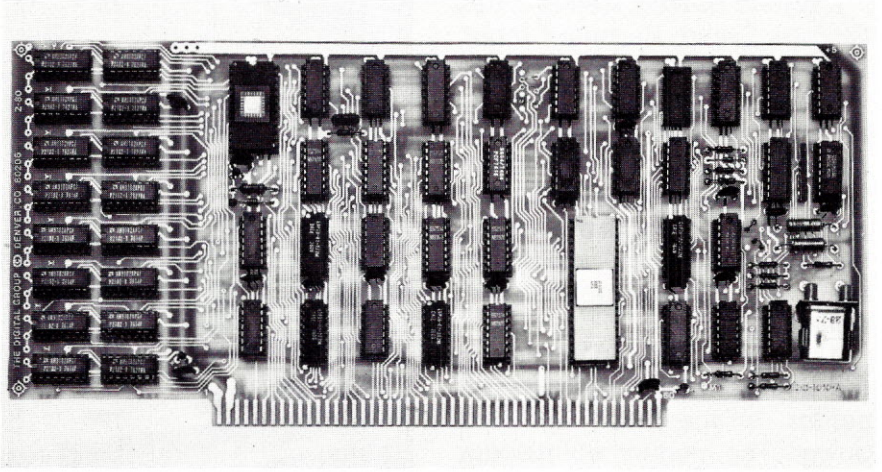


photo 6. The EPROM bootstrap loader and the 2K of RAM are on the left side of this Z-80 CPU board. Great care must be used in assembling this card due to the close proximity of the conductors. The external interrupt socket is the empty one near the top of the board on the right.

Z-80 CPU card

EPROM bootstap loading on this board provides you with 'power-on smarts' to eliminate the need for toggling in this information from a front panel. DMA circuitry makes it possible for you to add a front panel in case of emergency debugging requirements. 2K of address strappable RAM is provided on this card to store the operating system which we will discuss shortly.

This is a good place to mention one of the characteristic features of the DG system; processor independent system design. If you are concerned about newer and better microprocessor chips appearing on the market and what effect this will have on your investment, the DG system is for you. DG has an ongoing program of offering the latest and most popular processor chips designed onto cards which can be used to directly replace your present DG CPU. In fact, they have offered an 8080-to-Z-80 upgrading kit so you could strip the expensive parts from your 8080 board, buy the upgrade kit containing the Z-80 and other parts, put it together and *presto* — your mild mannered 8080 board has turned into the powerful Z-80 board, able to leap tall instructions in a single bound! As you can see, the DG is wise when it comes to spending your money.

Take your time constructing this board. It took me 9 hours of careful assembly and checking with a voltmeter for proper DIP socket voltages before I went for the 'big switch' and the moment of truth. Data paths are dense as you can see from **photo. 6.** My experience with computer kits has instilled in me a great respect for the care which *must* be exercised during the assembly of CPU boards such as this where the high density data paths seem to attract solder splashes and wayward strands of wire. Pretend that you are assembling the main board for a NASA Mars probe — it might help.

A calibrated scope is required to confirm that the clock is oscillating at exactly 2.5 MHz. Even a slight error here will be amplified many times by software timing loops. Program a time-of-day clock to check this frequency if you do not have access to a quality scope.

DG software support

The DG has a professional software group to generate programming support. Their major object, I would say, is user convenience. This systems approach is embodied in the Operating System Cassette which transforms the collection of hardware into a real 'computer.'

Cassette Read, Cassette Write, Storage Dump, Keyboard Input, and the Operations Monitor are the five programs which comprise the operating system. DG cautions the user to stay above location 006 000 (octal) unless you wish to overwrite the operating system (if you do, merely reload the Operating System Cassette).

After thoroughly DC checking the system and inspecting for solder bridges, turn on the power. The monitor will display, "READ Z-80 INITIALIZE CASSETTE." Load the operating system from a cassette recorder and observe the screen. If a "." shows up instead of the octal page number of the data being read in, your system has determined that a bit of data has been read incorrectly. Reload and start again. I would recommend that you modify the remote speaker jack on your cassette recorder so that you can hear the data as it flows into the interface. This will permit you to locate the operating system beginning and end, as well as that of other programs on tape.

The user options will be displayed on the screen as shown in **photo. 7.** The jump in the option numbers is due to a program improvement made by the DG software people. READ and WRITE are the cassette operating programs and are used as you might expect. Options 3 and 4 are the DUMP program, 3 providing an octal dump, 4 a hex dump. For example, select option 3 (via the keyboard) and the registers and flags of the Z-80 will instantly appear on the screen. Press the Space key to page through memory, 96 bytes at a time (it's just like flipping the pages of a book). This memory scanning feature makes it easy to locate unused memory for new programs.

photo 7. Although I did not buy this fancy monitor offered by the DG, this photo does show the clarity of the readout and the soft ware options posed by the operating system.

BASIC, games, etc.

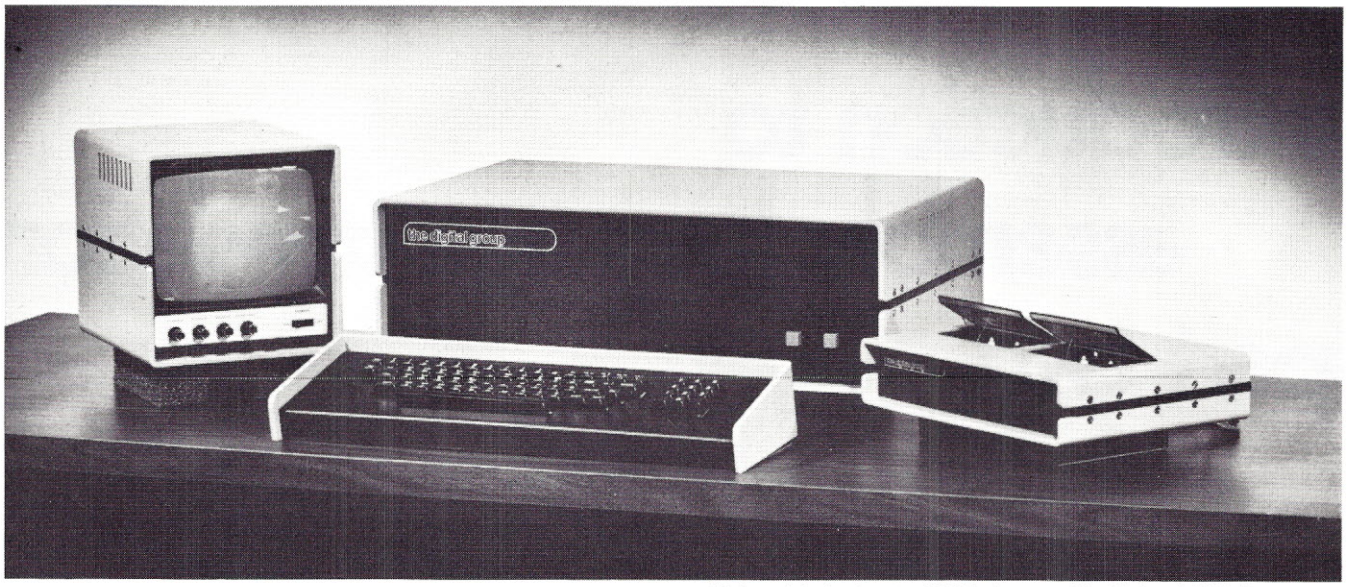
MAXI-BASIC is the high-level programming language offered by the DG. It contains all the features of standard BASIC and then some. This version of BASIC occupies 8K and the DG recommends that you have at least 18K to run it. MAXI-BASIC is available on a cassette from the DG for $15.

And speaking of BASIC, Tiny BASIC Extended (TBX-Z80) is offered for $5, also on a cassette. Why, you ask, should you buy an abbreviated version of BASIC when you can get MAXI-BASIC to feed your Z-80? Well, the DG collection of games on cassettes must run under TBX-Z80, that's why. So, if you're a gamesman (gamesperson?), be sure to get the TBX-Z80. The DG offers not just one, but six Tiny BASIC Game Cassettes. See the DG Flyer No. 8 for the full games listings. By the way, the CLOCK program on Set 3 has a problem — the timing loops were not completely adjusted to the Z-80 speed (the games were adapted from their 8080 versions) — don't set your watch by the CLOCK. For Z-80 ignoramuses, there is the Z-80 Educator Cassette. This program will quickly teach you about the Z-80 instructions and addressing modes.

Back to the Operating System Cassette, we find six neat little routines on the tape after the operating system. There is the CW KEYBOARD routine — outputs keying signals for a code generator on the LSB of port 2. It has a software FIFO which permits the operator to get up to 256 characters ahead of the character being sent. The RTTY-TO-ASCII routine converts FSK at 60 wpm (Baudot) to ASCII. BRAIN TEASER is a game for stretching your gray matter. Use FREQUENCY COUNTER to align your cassette interface, if need be. STAR SPANGLED BANNER was the routine mentioned at the beginning of this article. Finally, we get to MEMORY TESTER. Use this program to check out your 8K RAM boards. It prints out (are you ready) the IC number of any memory chip which it finds to be faulty. Not only that, but this program automatically 'knows' how much memory you have in your system. Now that's convenience.

conclusion

The Digital Group offers a very complete, conservatively designed system. The quality of the hardware, I have found, is excellent. Perhaps the most outstanding features are the processor independent system design to cope with advances in the technology (or applications requirements) and the user convenience of their software. From their delivery, sales, and repair policies, I find that they are reliable and responsive, and that's important. If that's what you're looking for (and you should be), then perhaps it's time to give the DG a call and request their complete flyer package to see if you agree with my assessments. END



Cabinets clockwise from top: CPU, Dual-cassette drive, Keyboard, 9" Monitor.

Meet The Digital Group

If you are seriously considering the purchase of a microcomputer system for personal or business use... or just beginning to feel the first twinges of interest in a fascinating hobby... the Digital Group is a company you should get acquainted with.

For many months now, we've been feverishly (and rather quietly) at work on our unique, high-quality product—a microcomputer system designed from the inside out to be the most comprehensive, easy-to-use and adaptable system you'll find anywhere. And our reputation has been getting around *fast*. In fact, you may have already heard a little something about us from a friend. We've found our own best salesmen are our many satisfied customers.

There's a good reason. Simply, the Digital Group has a lot to offer: state-of-the-art designs, a totally complete systems philosophy, unexcelled quality, reasonable software, affordable prices and the promise that our products will not become rapidly obsolete, even in this fast-moving, high-technology field.

The Advantages

Here are a few specific advantages of our product line:

- We offer interchangeable CPUs from different manufacturers (including the new "super chip"—the Z-80 from Zilog) which are interchangeable at the CPU card level. That way, your system won't become instantly obsolete with each new design breakthrough. The major portion of your investment in memory and I/O is protected.
- Digital Group systems are complete and fully featured, so there's no need to purchase bits and pieces from different manufacturers. We have everything you need, but almost any other equipment can be easily supported, too, thanks to the universal nature of our systems.
- Our systems are specifically designed to be easy to use. With our combination of TV, keyboard, and cassette recorder, you have a system that is quick, quiet, and inexpensive. To get going merely power on, load cassette and go!
- Design shortcuts have been avoided—all CPUs run at full maximum rated speed.
- All system components are available with our beautiful new custom cabinets. And every new product will maintain the same unmistakable Digital Group image.

The Features

Digital Group Systems—CPUs currently being delivered: Z-80 by Zilog 8080A/9080A 6800 6500 by MOS Technology

All are completely interchangeable at the CPU card level. Standard features with all systems:

- Video-based operating system
- Video/Cassette Interface Card
  512 character upper & lower case video interface
  100 character/second audio cassette interface
- CPU Card
  2K RAM, Direct Memory Access (DMA)
  Vectored Interrupts (up to 128)
  256 byte 1702A bootstrap loader
  All buffering, CPU dependencies, and housekeeping circuitry
- Input/Output Card
  Four 8-bit parallel input ports
  Four 8-bit parallel output ports
- Motherboard

Prices for standard systems including the above features start at $475 for Z-80, $425 for 8080 or 6800, $375 for 6500.

More

Many options, peripherals, expansion capabilities and accessories are already available. They include rapid computer-controlled cassette drives for mass storage, memory, I/O, monitors, prom boards, multiple power supplies, prototyping cards and others. Software packages include BASICs, Assemblers, games, ham radio applications, software training cassettes, system packages and more (even biorhythm).

Sounds neat—now what?

Now that you know a little about who we are and what we're doing, we need to know more about you. In order for us to get more information to you, please take a few seconds and fill in our mailing list coupon. We think you'll be pleased with what you get back.

---

the digital group

P.O. Box 6528
Denver, Colorado 80206
(303) 777-7133

OK, I'd like to get to know you guys better.
Send me the whole package!

Name _____

Address _____

City/State/Zip _____

introduction to assembly language translation part two

by Michael Wimble
6026 Underwood Ave., S.W.
Cedar Rapids, Iowa 52404

In the first part of this short course on assembly language translation we discussed the organization and general features of a three-pass assembly language translator. Continuing with the discussion, this article examines phase one of the translator. Topics include: statement scanning, label processing, intermediate text generation, building and organizing a symbol table, and op code and operand processing. The translator considered here was written for the IBM System III computer. This assembler provides a wide range of user options. We will consider restrictions imposed to match this translator to systems which have a major resource limitation, such as a small main memory in the typical microcomputer.

phase one overview

As stated in part one of this series, the translator converts source programs (computer instructions written by the programmer) into strings of numbers which have precise meanings and are convenient to manipulate during the translation process. These strings of numbers are called *intermediate text.* An overview of the process used to generate intermediate text is shown in **fig. 1.** As shown in this flowchart, phase one of the translator performs the following for each source statement encountered in the program: the label attached to the statement is processed (just how this is done will be covered later) if any label is present; the op code (operation code) is extracted (this process will also be considered later), and for each particular op code there is a unique subroutine which is called to process the remainder of the source statement. The intermediate text produced during this phase must accurately express both the syntax (statement structure) and the semantics (precise meaning) of the source statements. In addition, intermediate text must contain several items of information describing the relationship of each source statement to the rest of the program.

Keeping this simple view of phase one in mind, we now turn to consider the content and format of the intermediate text to be generated by phase one. Later, the basic elements of information, called *tokens,* which are used to generate the intermediate text will be discussed. The examination of phase one of an assembly language translator will conclude with a discussion of the subroutines and tables which comprise an important part of the translation process.

intermediate text

Each source statement is scanned by a subroutine which is appropriately called SCAN. This routine recognizes and extracts the semantic elements (the identifiable units of information) encountered in each source statement. The extracted semantic elements are then converted into intermediate text (number strings) containing the following information:

1. The statement number
2. The statement label (if present)
3. The value of the location counter when the statement was found
4. The op code
5. The length (in bytes) of the code to be generated by the translation
6. The operand expression in postfix notation (as discussed in Part One)
7. A description of the format of the operand
8. Error information

fig. 1. General flow of phase one showing the relationship of major subroutines.

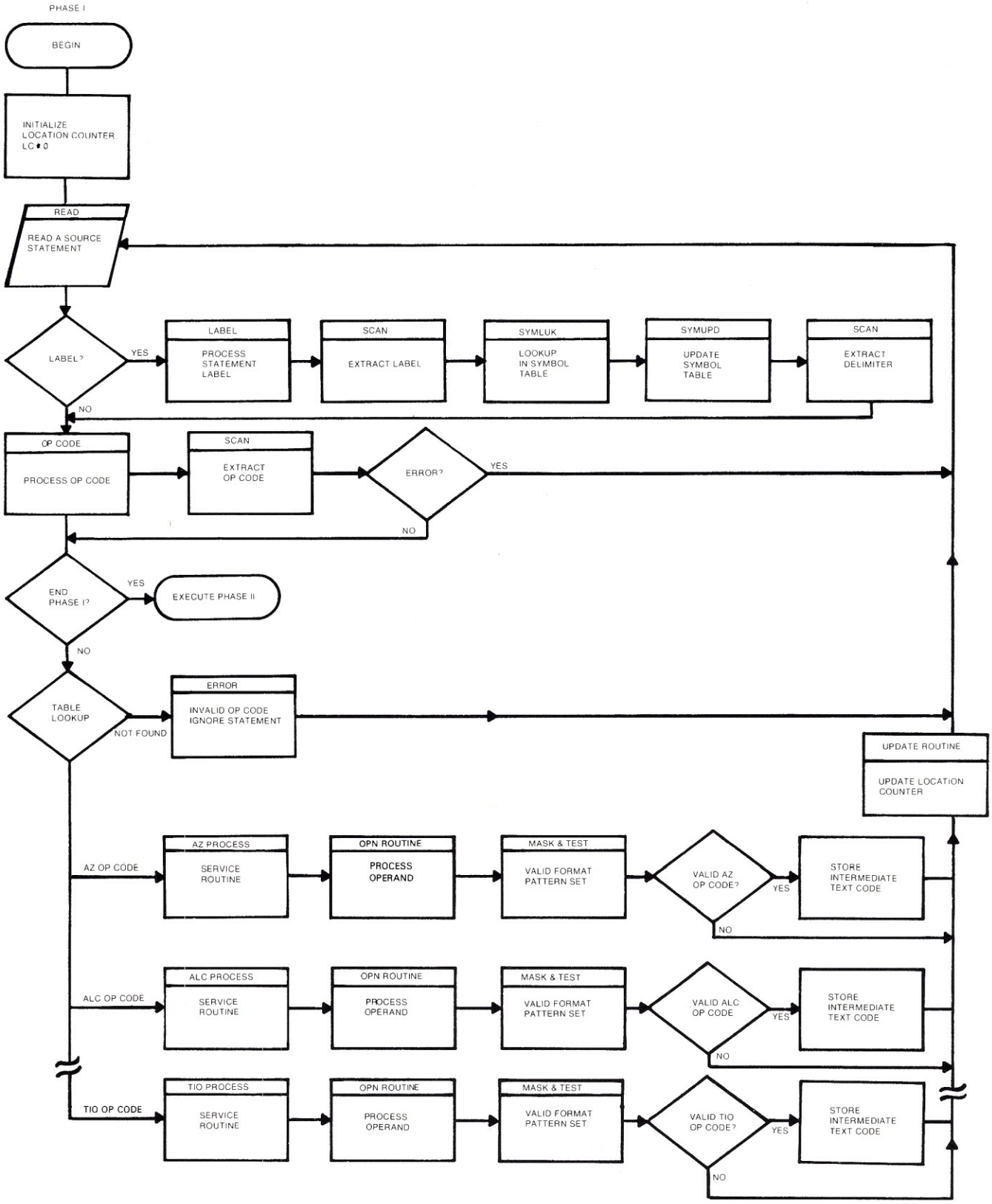


table 1. Program verbs are encoded using this Hex chart. Below are the definitions of these verbs.

VERB HEX CODING TABLE

1ST HEX DIGIT → / 2ND HEX DIGIT ↓	0	1	2	3	8
0	FILL	SPACE	YABS	IF	*CSECT
1	LTRM	EJECT	XABS	THEN	*ASECT
2	ORG	TITLE	REC#	ELSE	*USECT
3	DROP	EQU	PRPA	*ARG	*DATA
4	USING	LOCC	PLPA	*LENGTH	*DO
5	TEXTC	MASK	*TLS	ABS	*ELSE2
6	TEXT	LABEL	SFT	*NARG	*FIN
7	DBS	QB	MOD	*COM	*DO1
8	DBC	ALEN	NEQ	*NCOM	*GEN
9	DS	ALOC	LE	NUMBER	D1
A	DC	ERR	GE	ELPA	D2
B	END	OPND	XOR	UNEG	L1
C	EXTRN	ACOD	UNDEF	*SLPA	L2
D	ENTRY	RCOD	SYM	ARGS	X1
E	START	FABS	*TEST	RTRM	X2
F	PRINT	ZABS	*CNT	*SRPA	*CERR

* => UNIMPLEMENTED EXTENSIONS

MNEMONIC	DEFINITION
FILL	FILL CHARACTER. USED TO PAD TEXT IF NECESSARY, IGNORED.
LTRM	LEFT TERMINATOR. FIRST BYTE OF TEXT, SIGNALS BEGINNING.
ORG	ORIGIN DIRECTIVE SCANNED.
DROP	DROP BASE REGISTER DIRECTIVE SCANNED.
USING	USING BASE REGISTER DIRECTIVE SCANNED.
TEXTC	TEXTC OP CODE SCANNED. OPERAND FIELD WILL CONTAIN A TEXT STRING.
TEXT	TEXT OP CODE SCANNED. OPERAND FIELD WILL CONTAIN A TEXT STRING.
DBS	DEFINE STORAGE. LABEL ASSIGNED LOCATION OF 1ST BYTE RESERVED.
DBC	DEFINE CONSTANT. LABEL ASSIGNED LOCATION OF 1ST BYTE RESERVED
DS	DEFINE STORAGE. LABEL ASSIGNED LOCATION OF LAST BYTE RESERVED.
DC	DEFINE CONSTANT. LABEL ASSIGNED LOCATION OF LAST BYTE RESERVED.
END	END DIRECTIVE SCANNED.
EXTRN	EXTRN DIRECTIVE SCANNED.
ENTRY	ENTRY DIRECTIVE SCANNED.
START	START DIRECTIVE SCANNED.
PRINT	PRINT DIRECTIVE SCANNED.
SPACE	SPACE DIRECTIVE SCANNED.
EJECT	EJECT DIRECTIVE SCANNED.
TITLE	TITLE DIRECTIVE SCANNED.
EQU	EQU DIRECTIVE SCANNED.
LOCC	LOCATION COUNTER REFERENCE IN EXPRESSION.
MASK	MASK FOLLOWS IN NEXT TWO BYTES. THE MASK CONTAINS SWITCHES INDICATING WHICH OF THE POSSIBLE PARTS OF AN OPERAND ARE ENCOUNTERED. FOR THIS TRANSLATOR AN OPERAND CAN OCCUR IN THE FOLLOWING FORMAT: D1(L1,X1),D2(L2,X2) AND EACH BIT IN MASK DENOTES ONE OF THE POSSIBLE ELEMENTS OF THE FORMAT FOUND.
LABEL	SYMBOL NUMBER OF LABEL FOR THIS STATEMENT FOLLOWS IN NEXT TWO BYTES.
QB	Q-BYTE OF INSTRUCTION FOLLOWS IN NEXT BYTE.
ALEN	LENGTH OF ASSEMBLED CODE FOR THIS STATEMENT FOLLOWS IN NEXT TWO BYTES.
ALOC	VALUE OF LOCATION COUNTER FOR STATEMENT FOLLOWS IN NEXT TWO BYTES
ERR	ERROR DEFINITION FOLLOWS IN NEXT TWO BYTES. 1ST BYTE IS COLUMN NUMBER SCANNED WHEN ERROR OCCURED, 2ND BYTE IS ERROR NUMBER.
OPND	OPERAND EXPRESSION FOLLOWS. 1ST BYTE FOLLOWING IS LENGTH OF OPERAND. EXPRESSION STRING FOLLOWS
ACOD	ABSOLUTE DATA FOLLOWS. 1ST BYTE IS LENGTH OF DATA STRING. DATA STRING FOLLOWS.
RCOD	RELOCATABLE DATA FOLLOWS. 1ST BYTE IS LENGTH OF DATA STRING. DATA STRING FOLLOWS.
FABS	FORMAT 'F' OP CODE FOLLOWS IN NEXT BYTE.
ZABS	FORMAT 'Z' OP CODE FOLLOWS IN NEXT BYTE.
YABS	FORMAT 'Y' OP CODE FOLLOWS IN NEXT BYTE.
XABS	FORMAT 'X' OP CODE FOLLOWS IN NEXT BYTE.
REC#	STATEMENT NUMBER FOLLOWS IN NEXT TWO BYTES.
PRPA	RIGHT PARENTHESIS FOLLOWING X1 OR X2 SCANNED.
PLPA	LEFT PARENTHESIS PRECEDING L1 OR L2 SCANNED.
TLS	UNDEFINED
SFT	SHIFT OPERATOR SCANNED. ('**')
MOD	MODULUS DIVIDE OPERATOR SCANNED. ('//')
NEQ	NOT EQUAL OPERATOR SCANNED. ('¬=')
LE	LESS THAN OR EQUAL OPERATOR SCANNED. ('<=')
GE	GREATER THAN OR EQUAL OPERATOR SCANNED. ('>=')
XOR	EXCLUSIVE OR OPERATOR SCANNED. ('||')
UNDEF	UNDEFINED SYMBOL ENCOUNTERED.
SYM	SYMBOL # OF SYMBOL IN EXPRESSION FOLLOWS IN NEXT TWO BYTES.
TEST	UNDEFINED
CNT	UNDEFINED
IF	IF OPERATOR SCANNED. ('"F')
THEN	THEN OPERATOR SCANNED. ('"N')
ELSE	ELSE OPERATOR SCANNED. ('"E')
ARG	UNDEFINED
LENGTH	UNDEFINED
ABS	ABSOLUTE VALUE OPERATOR SCANNED. ('"A')
NARG	UNDEFINED
COM	UNDEFINED
NCOM	UNDEFINED
NUMBER	ABSOLUTE NUMBER IN EXPRESSION FOLLOWS IN NEXT TWO BYTES.
ELPA	EXPRESSION LEFT PARENTHESIS. (APPEARS ONLY IN STACK)
UNEG	UNARY NEGATIVE OPERATOR SCANNED. ('-')
SLPA	UNDEFINED
ARGS	ARGUMENT SEPARATING COMMA SCANNED.
RTRM	RIGHT TERMINATOR. MARKS END OF INTERMEDIATE TEXT
D1	D1 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
D2	D2 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
L1	L1 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
L2	L2 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
X1	X1 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
X2	X2 EXPRESSION FOLLOWS IN NEXT 3 BYTES. 1ST BYTE IS ZERO FOR ABSOLUTE EXPRESSION. NON ZERO FOR RELOCATABLE. BYTES 2-3 ARE VALUE.
CERR	UNDEFINED

Table 1 shows the Hex coding plan used to translate program verbs into intermediate text. **Fig. 2** provides an example which demonstrates the format and content of the intermediate text produced for two statements encountered in a source program. The 'X, Y, Z' formats specify the arrangement of elements used to form the op code.

It should be clear that intermediate text is produced by scanning the source statement, extracting the semantic elements one by one, converting them into internally useable number strings, reordering these numbers, and adding descriptive information such as statement numbers and location counter values. The next section describes the subroutine responsible for scanning the source statement, extracting the semantic elements, and partially converting these into intermediate text.

SCAN-token extraction

A token is the basic unit of semantic information in any programming language statement. Any combination of characters may constitute a token, while combinations of tokens constitute a statement, and combinations of statements constitute a program. The bulk of programming language translation consists of manipulating tokens. As a result, a subroutine called SCAN has been developed to extract tokens from source statements.

As each source statement is read, pointers are initialized so that SCAN begins searching at the first character of the record. Each call to SCAN (i.e., "CALL SCAN") results in the extraction of the next semantic unit of information. This extracted element is stored in a buffer memory and flag bigs are set to describe what has been found

by SCAN. This descriptive information is *returned* to the calling program to aid in the translation process.

tokens

Tokens fall into one of two broadly defined classes: identifiers and operators. Operators are, of course, symbolic mathematical operators (+, /, etc.). Operators also include delimiters (blanks, ( ), etc.). Identifiers are then tokens which are not operators. Identifiers consist of symbols (A, 16XZM, etc.) and numbers (−1, 36287, etc.). A token is *terminated* by the occurance of an operator. **Table 2** shows the single character operators for the sample translator.

To illustrate the operation of subroutine SCAN, **fig. 3** shows the flowchart for SCAN and **tables 3 and 4** list the communication variables and bit switches used by SCAN. **Fig. 4** illustrates the operation of SCAN as tokens are extracted from a sample source statement.

communication bits

As shown in **table 4,** there are 8 communication bits which can be set by SCAN to describe the token which it has found. The calling routine must test for the various conditions indicated by these communication bits on return from SCAN. If SYMF (SYMbol Found) is set, then either a symbol or a number was scanned; if *both* ALFA ("ALFAbetic") and SPEC (SPECial) are reset, a number was scanned; otherwise a symbol was scanned. If a number was found, the first three bytes of SCTOK (SCan TOKen) contain the 24 bit representation of the number. A scanned symbol is stored left justified in the correct character format beginning at SCTOK.

If LITS (LITeral Scan) is set a literal was scanned. The literal is stored in character format, left

fig. 2. Example of intermediate text generated for two typical source statements.

```
STMT  LABEL    STATEMENT
50    LOOP1    AZ SUM(5),HOURS(3*LEN,XR1)
51             LA 1(,XR1),XR1
```

TEXT (HEX)	MEANING
22 ØØ 32	Statement No. 5Ø
2D ØØ 11	Label is symbol No. 17 (LOOP1)
19 Ø4 0C	Location counter value is Ø4ØC (HEX)
21 Ø6	The 'X format' op code is Ø6 (AZ)
18 ØØ Ø5	Length of code to be generated is 5 bytes
15 E7 F8	Standard operand form is: D1(L1),D2(L2,X2)
1B 15	Operand expression is contained in the next 21 bytes
2D ØØ 12	Symbol is No. 18 (SUM)
24	Left parenthesis: (
39 ØØ Ø5	The number 5
23	)
3D	, argument separator
2D ØØ 31	Symbol No. 49 (HOURS)
24	(
39 ØØ Ø3	The number 3
2D ØØ 13	Symbol No. 19 (LEN)
5C	* (an EBCDIC character)
6B	, (" " ")
2D ØØ 14	Symbol is No. 2Ø (XR1)
22 ØØ 33	New statement No. is 51
19 Ø4 11	Location counter value is Ø411 (HEX)
1F Ø2	The 'Z format' op code is Ø2 (LA)
18 ØØ Ø3	Length of code to be generated is 3 bytes
15 9F ØØ	Standard operand form is: D1(,X1),D2
18 ØD	Operand expression follows in the next 13 bytes
39 ØØ Ø1	The number 1
24	(
3D	, argument separator
2D ØØ 14	Symbol No. 2Ø (XR1)
23	)
6B	, (an EBCDIC character)
2D ØØ 14	Symbol No. 2Ø (XR1)

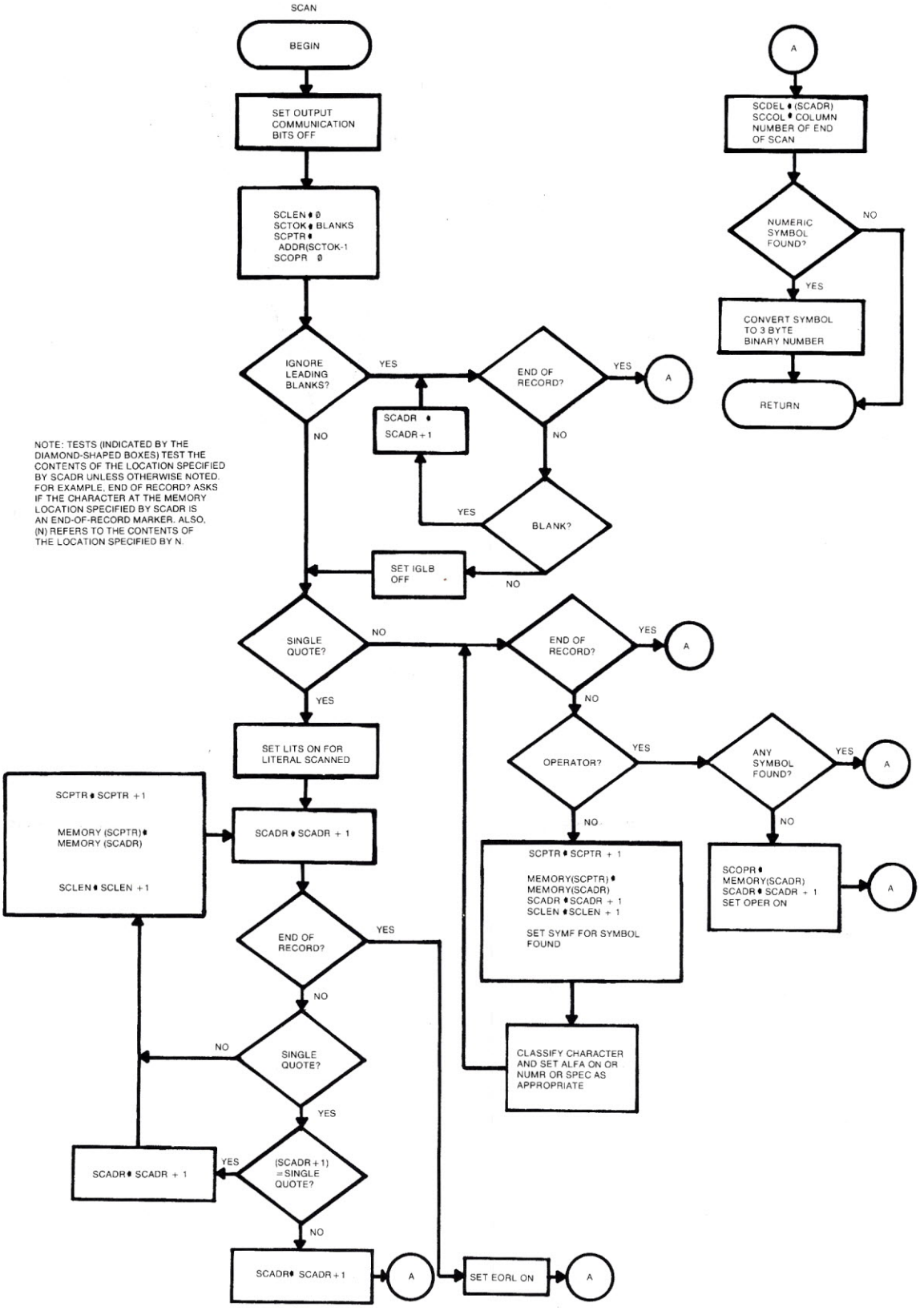


fig. 3. Flowchart of subroutine SCAN.

justified, beginning at SCTOK. When LITS is set, the assembler must test EORL (End Of Record Literal). EORL set means the closing quote for a literal was not found by the time the end of record was reached.

If neither LITS nor SYMF is set, one of two conditions has occurred: either an operator was scanned or the end of record was found. If an operator was scanned, OPER (OPERator) will be set.

```
LABEL     STATEMENT
LOOP1  AZ SUM(5),HOURS(3*LEN,XR1)

CALL  TOKEN   DELIMITER   OPERATOR
  1   LOOP1   blank
  2   AZ      blank
  3   SUM     (
  4           5           (
  5   5       )
  6           ,           )
  7           H           ,
  8   HOURS   (
  9           3           (
 10   3       *
 11           L           *
 12   LEN     ,
 13           X           ,
 14   XR1     )
 15           blank       )
```

fig. 4. Example of token extraction by subroutine SCAN.

IGLB switch

Before calling SCAN, the calling routine can set a bit called IGLB (IGnore Leading Blank). This condition causes SCAN to ignore leading blanks in any statement which it scans. The IGLB option is useful when trying to locate the beginning of the op code or operand fields. For instance, after having found the blank character which terminates the label field, the next call to SCAN should have set IGLB so that any blanks between the label and the op code fields will be ignored. The symbol returned will then be just the desired op code. Without this facility, the calling program would have to make repeated

table 2. Single character operators used in the sample translator.

CHARACTER	NAME	FUNCTION
,	comma	label/argument/operand separator
'	single quote	enclosing literals
+	plus	binary sum/unary positive value
—	dash	binary subtract/unary negative
¬	not	logical inverse
*	asterisk	location counter/binary multiply
/	slash	binary divide operator
(	left parenthesis	expression grouping/sub-parameter definition (index register)
)	right parenthesis	closure of left parenthesis
<	less than	relational binary operator
>	greater than	relational binary operator
=	equal	relational binary operator
|	vertical bar	logical binary operator OR
&	ampersand	logical binary operator AND

table 3. Communication variables used by subroutine SCAN.

VARIABLE NAME	FUNCTION
SCADR	Holds memory address of next character to be scanned
SCCOL	Holds column number of end of scan (one column before delimiter)
SCDEL	Holds delimiter encountered
SCLEN	Holds length of symbol, number, or literal in SCTOK
SCOPR	Holds operator scanned
SCPTR	Holds memory address of last character put into SCTOK
SCTOK	Holds extracted symbol, number, or literal

*table 4. Communication bits (also called **switches**) used by SCAN.*

BIT NAME	BIT	INPUT/ OUTPUT	FUNCTION
IGLB	Ø	INPUT	ON means ignore leading blanks
LITS	1	OUTPUT	ON means literal was scanned
SYMF	2	OUTPUT	ON means symbol was scanned
ALFA	3	OUTPUT	ON means alphabetic character was scanned
NUMR	4	OUTPUT	ON means numeric character was scanned
SPEC	5	OUTPUT	ON means special character was scanned
OPER	6	OUTPUT	ON means an operator was scanned
EORL	7	OUTPUT	ON means end of record was found while scanning a literal

calls to SCAN and examine the communication variables to determine when a symbol was finally found.

Having now defined a token, described its extraction and shown its ultimate use, we may continue with the discussion of the translator itself. The next section will detail label processing. Following this there will be a brief digression to discuss symbol tables before returning to the translation process.

LABEL subroutine

Fig. 5 shows the flowchart for subroutine LABEL. There is a feature implied here which is not found in many assemblers; the ability to assign multiple labels to a statement. This is useful for writing large programs, such as language translators, and does not cost much in terms of translator size or complexity. With the multiple label facility, several assignments can be accomplished with a single statement. For instance, to equate the three symbols "PARMEG", "R1", and "ONE" equal to a numerical value of 1 we can use a single statement:

PARMREG, R1, ONE EQU 1

instead of three statements as would be required by most other assemblers.

Following along **fig. 5,** the first step is to call SCAN (IGLB not set) to extract any symbol starting in column 1 of the record. If a literal or an operator is found, then error information is generated in the intermediate text string and the statement is ignored. If no symbol was found, a check is made for the blank character which terminates the label field. Absence of this blank indicates invalid statement syntax.

Having extracted the label, the next step is to call SYMLUK which performs two functions and is described in detail in the next section. First a check is

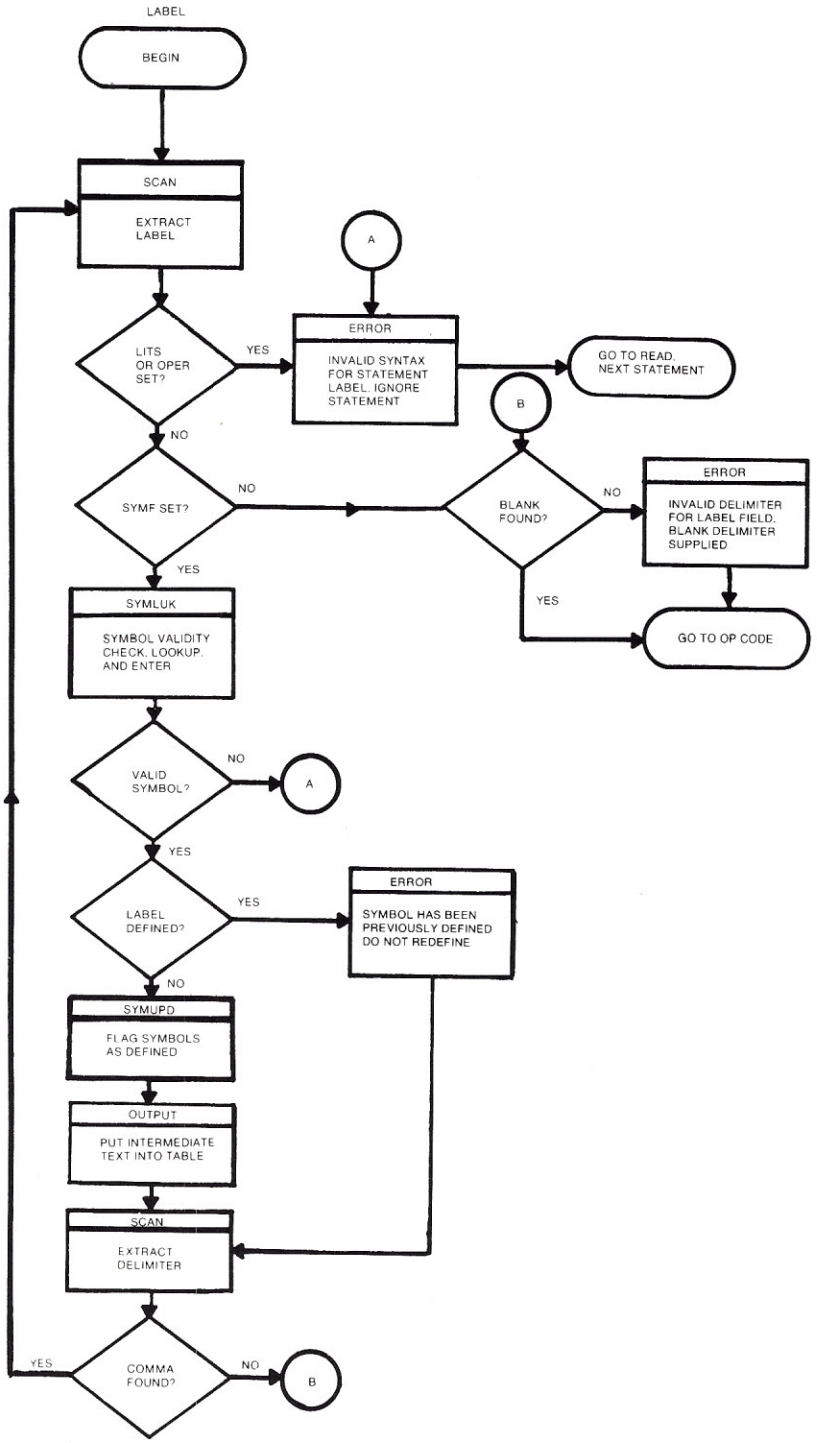


fig. 5. Flowchart of subroutine LABEL.

made to determine valid symbol syntax. If a valid symbol is found then a lookup is made in the symbol table to see if the symbol has already been entered. If the symbol is already in the table then it must have been previously found in an operand expression or as a label of a previous statement. A symbol appearing in a label field is said to be defined, so a symbol appearing as a label more than once would be multiply defined, which is an error. In either case, when a symbol is found in a label field its symbol table entry is updated to reflect that the symbol is defined. Symbols not already in the symbol table are added and flagged as defined.

After creating the intermediate text for any valid labels encountered, SCAN is invoked again to extract the label's delimiter. If a comma is found, the whole process is repeated, otherwise the previously described check for blank delimiter is performed.

Before proceeding with the description of the next step in the translation function, which is to extract and service the opcode, another digression will be made. The next section discusses the organization, content, and manipulation of the symbol table; important and necessary background material.

symbol table organization

The symbol table is perhaps the most important single element of a language translator. Symbolic manipulation is responsible for raising the level of programming from the tedium of machine language programming to the ease of high level language programming. Indeed, the power and popularity of a language is directly related to the facilities for symbol manipulation provided by the language.

In order to increase the utility of a symbol table, more information must be maintained, which

BYTE	BIT	FUNCTION
Ø - 5		Character representation of the symbol
6 - 7		Symbol value
8 - 9		Statement No. where symbol is defined
10 - 11		Implied length of symbol
12		Other descriptive information as follows
	Ø	EXTRN symbol
	1	ENTRY symbol
	2	Multiply defined
	3	Value is resolved
	4	Value is absolute
	5	Symbol is name of module
	6	Macro switches in byte 15 apply
	7	Definition has been encountered
13 - 14		Hash forward link (also XREF link)
15		Macro processing switches

fig. 6a. Byte definition of each of the 16 bytes of information required by the sample translator for every symbol table entry.

BYTE	BIT	FUNCTION
Ø - 4		Character representation of the symbol
5 - 6		Value of the symbol
7		Other descriptive information:
	Ø	Multiply defined
	1	Value is resolved
	2	Value is absolute
	3	Definition has been found
	4	EXTRN or REF symbol
	5	ENTRY or DEF symbol
	6	Symbol is module name
	7	Available

fig. 6b. Byte definition for a minimum size symbol table requiring only 8 bytes of information for each table entry.

table 5. Communication variables used by symbol table subroutines.

VARIABLE NAME	FUNCTION
SYEND	Number of symbols currently in table
SYMAX	Maximum number of symbols that can be stored in symbol table
SYNUM	Number of symbols found or to be updated
SYSYM	Copy of symbol found, to be created, or to be updated

table 6. Communication bits used by symbol table subroutines.

BIT NAME	BIT	FUNCTION
VNUM	Ø	ON means symbol is a valid number
VALF	1	ON means symbol is a valid alphanumeric
FSYM	2	ON means symbol was found in the symbol table
NSYM	3	ON means this is a new symbol

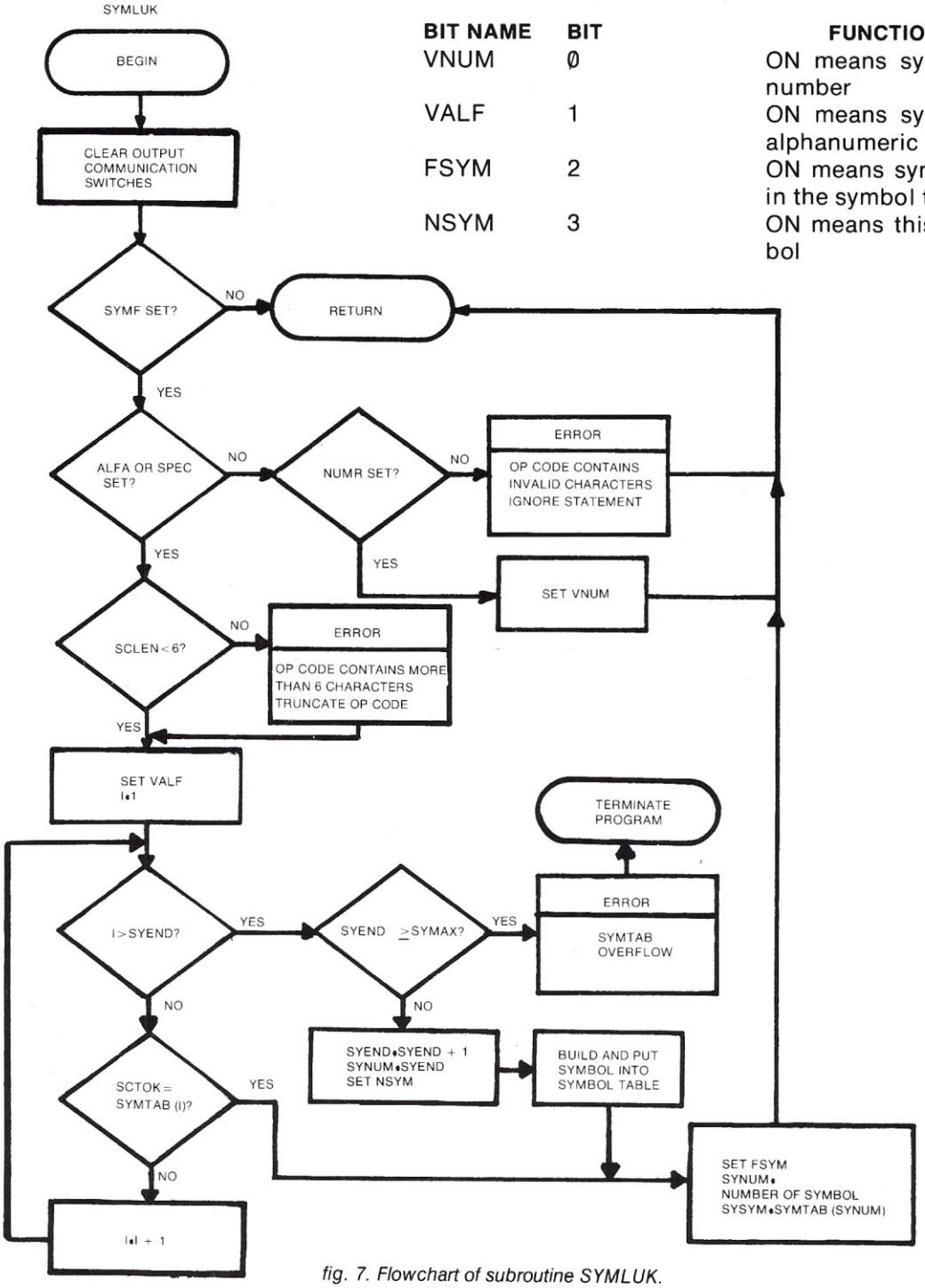


fig. 7. Flowchart of subroutine SYMLUK.

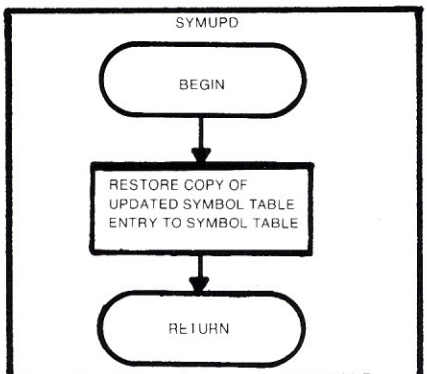


fig. 8. Flowchart of SYMUPD.

increases both the size and complexity of the table. This usually necessitates restricting the number of symbols that the translator can process in a single run. By reducing the amount of information maintained in the symbol table, more symbols may be processed at a cost of trading off many powerful features and conveniences, such as intelligent error recovery and macro processing.

As shown in **fig. 6a.**, each entry into the symbol table used by the sample translator requires 16 bytes of information. **Fig. 6b.** shows the minimum information needed by a useful translator.

The organization of a symbol table is at least as important as its content. The simplest organization is a *linear list* where new elements are added to the table immediately following the previous entries. To perform a symbol table lookup, that is to check if a symbol is already entered into the table, a comparison is made between the symbol to be found with each entry in the table until a match is determined or until the end of the table is reached. The single advantage of the linear list's simplicity must be weighed against the inordinate amount of time required to perform a lookup in this type of table.

In phase one of the translator, symbol table lookup is a very frequent activity. Storing a symbol table as a linear list instead of in a more sophisticated manner could mean the difference

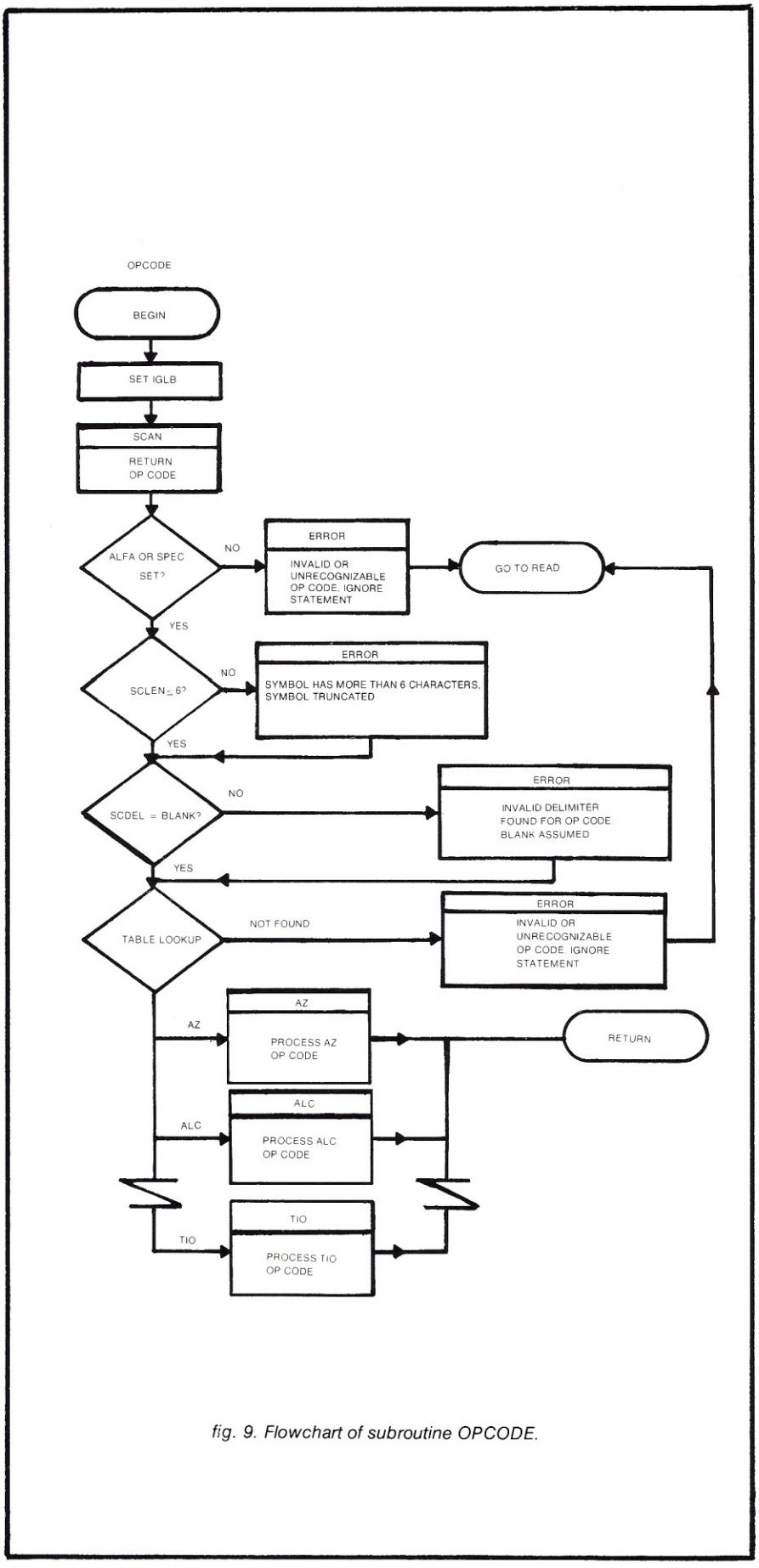


fig. 9. Flowchart of subroutine OPCODE.

MASK BIT	MEANING OF BIT SET
Ø	D1 found in operand
1	(after D1 found in operand
2	L1 found in operand
3	, between L1 and X1 found
4	X1 found in operand
5	) after X1 found in operand
6	, between operand subexpressions found in operand
7	D2 found in operand
8	(after D2 found in operand
9	L2 found in operand
A	, between L2 and X2 found
B	X2 found in operand
C	) after X2 found in operand
D	- not used -
E	- not used -
F	Operand could not be matched to valid pattern

fig. 10. Definition of bits in the 16 bit mask (pattern) used to examine (compare) with encoded text to determine whether the text follows the proper format.

between a 30 minute translation versus a 30 second translation. For the purpose of illustration, the flowcharts in **fig. 7** and **fig. 8** assume that the symbol table is organized as a linear list. However, the table used in the sample translator employs hash coding, linked lists, and overflow tables to achieve highly efficient table searching and storage. This technique is not suitable for computers with small main memory.

SYMLUK and SYUPD

Phase one performs three operations on the symbol table: lookup, add, and update. Lookup and add are performed by the same subroutine, SYMLUK, while update is performed by SYMUPD.

Fig. 7 shows the logic involved in SYMLUK and **tables 5 and 6** list the appropriate communication variables and bits. It is convenient to let SYMLUK also perform some validation checks on the identifier, thus SYMLUK begins by checking the communication switches from SCAN. If SYMF is not set, SYMLUK terminates since literals and operators are not a part of the symbol table.

If SYMF is set, meaning that an identifier of some sort was scanned, then ALFA and SPEC are checked. If one or both are set, then a valid symbol was found. Remember from Part One of this article that a valid symbol is any string of characters that can't be mistaken for something else. Well, if SYMF is set then the identifier scanned must be some combination of letters, digits, and special characters. If the identifier is composed solely of digits it must be a number, otherwise it is a valid symbol. Finding an identifier consisting solely of digits results in VNUM (Valid NUMber) being set while an identifier not containing any digits, alphabetic characters, or special characters is considered an error and error information is put into the intermediate text.

Having found a valid symbol, SYMLUK will only consider the first six characters significant, truncating longer symbols. VALF is set to indicate that the identifier is a valid symbol. Next, a table lookup is performed to determine if the symbol is already in the table. If the symbol is not found it is added. Finally FSYM is set to indicate that the symbol was found and a copy of the table entry is saved in SYSYM for use by the calling routine.

Fig. 8 shows the simple process of symbol table update. The symbol table entry stored in SYSYM by SYMLUK is updated by the calling routine which eventually invokes SYMUPD to replace SYSYM into the symbol table.

We return now to describing the phase one translation process. The next section describes the extraction and processing of the op code. Following this we get into what can be called the "meat" of the assembler — the extraction and encoding of operand expressions.

OPCODE subroutine

Op code extraction and processing is relatively simple and is shown in **fig. 9.** A call is made to SCAN with a request to ignore leading blanks. The identifier returned is checked for validity: proper size, character set, and delimiter. A table lookup is then performed to determine if the op code is valid. If valid, the table also indicates the address of the subroutine which will complete the processing of the source statement.

Until now, the only intermediate text that has been created for a statement describes the statement number, the symbol numbers of any labels encountered, and error information for any errors which may have been detected. It remains for each *service routine* to complete the intermediate text for the statement. Depending upon the op code, the service routine must generate text to describe the value of the location counter, the number of bytes of machine code to be produced as a result

Reader Input

PLEASED WITH FLOWCHARTING

I was pleased when I received my first issue of MICROTREK in the mail. I was even more pleased when I saw what MICROTREK contained. I was particularly impressed by your article on flowcharting. It has helped me in writing programs a great deal. I am also in the market to buy a microcomputer. After reading your excellent article on the KIM-1 Microcomputer Module, I seriously started to consider the purchase of such a unit.

Matthew A. Treu
Student,
Milwaukee Technical High School

PUBLIC SAFETY COMPUTERS

Although I am an electronics engineer, I have been involved in radio communications and therefore know little about computers. It is my hope, however, to correct this situation because of the tremendous impact computers are having upon public safety communications. Also, I would like to develop my own computer to work along with my interest (of some 35 years) in amateur radio. But, as I have indicated, I do not know enough at this time to ask a decent question in either hardware or software. So please have some very simple articles.

Erling R. Jacobsen, W9KBE
1438 34th Avenue
Rock Island, Illinois 61201

TERMINOLOGY BARRIERS

I hope that your publication continues to serve those without a background in computers. Most publications in this field assume that the reader is already in the field. Getting an understanding of the terminology used is a problem in itself.

Allan A. Simpson
17 Barberry Rd.
Winnipeg, Manitoba
Canada R2J 2G8

DON'T DROP OUT OF THE RACE!

Hope you guys have not decided to drop out of the micro-race. We have only just begun as they say. Down the road there is a new industry ahead. The reason I am saying this is that I haven't received a second issue on my subscription as yet and fear your demise.

Having once been a publisher of a small publication as well as the chief writer of it [*Have I got a job for you — Ed.*], I can readily appreciate the problems of starting up a publication. However, it is worth the struggle and I don't want to see you guys give up. You can always come out with a quarterly or heck yearly [*Some people claim that we have — Ed.*].

Maybe it isn't true that you are out of the micro-race. As Mark Twain said, "Rumors of my death are greatly exaggerated." [*I couldn't have put it better — please see merger details on page 29*].

Bill Haslacher
720 S. Third St.
San Jose, CA. 95112

MERGING MAGAZINES

As has been stated in the memo on page 6, and detailed in the merger plan on page 29, MICROTREK has merged with PERSONAL COMPUTING. Benwill Publications Corporation of Brookline, Mass. has had enormous success with PERSONAL COMPUTING and has invited MICROTREK to merge with them so that our readership can enjoy a broader spectrum of small computer applications as well as a magnitude of improvement in the quality and quantity of the graphical and editorial material. I apologize for the delay in our publication schedule which was entirely due to numerous problems experienced by our small MICROTREK staff. Now that MICROTREK is appearing in the PERSONAL COMPUTING magazine, I would like to ask you send me comments regarding our new "image" and specifically how you believe it can be improved. With the new facilities at our command, I know that you will enjoy the fruits of this merger. Please note that we now have an increased need for hardware, software, and applications articles. Send outlines or complete manuscripts to: PERSONAL COMPUTING, MICROTREK EDITOR, 167 Corey Road, Brookline, Mass. 02146.

Clubs and Groups

MACH is the one

If you're in the Omaha area and wish to exchange information on construction and software development, MACH may be your speed. For meeting details of MACH (Mid-America Computer Hobbyists), contact Lt. Tom Smith at 2708 Calhoun St., Bellvevue, Ne. 68005 or phone 292-6031.

computers French style

Ordinateur Libre is a French speaking computer group situated in Montreal. If the 75 member turn-out for the first meeting is any indication, Ordinateur Libre is going to be a popular group. Contact Real Menard, 2793 Dumont, Mascouche Prov. Que., Canada JON 1C0 for meeting details.

SMUG statistics

The SMUG (Sacramento Microcomputer Users Group) reports in their newsletter, *Push & Pop* that 60% of their members (or more accurately, 60% of their survey respondees) said they have some type of computer. Average memory (RAM) owned was 15K bytes. However, when polled to determine the membership interests, over 50% of the respondees checked a box labeled, "What are the little buggers, hardware, software." Hmmm. For meeting details contact Richard J. Lerseth at SMUG, P.O. Box 161513, Sacramento, CA. 95816. Meeting details will also be published regularly in *ON LINE.*

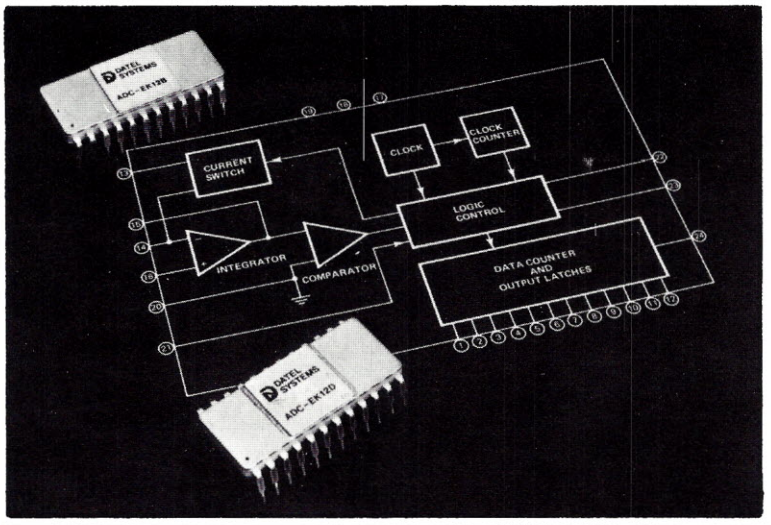
Panhandlers Society

A new computer hobby club, less than 3 months old, has been formed for interested computerists in the Texas Panhandle. More than 30 members meet every Friday and meetings are open to anyone interested in computers. The Panhandle Computer Society welcomes communication with other computer groups, especially in Texas. All inquiries should be addressed to Tex Everett or Jerry Fewell at 2923 S. Spring, Amarillo, Texas 79103 or call (806) 373-8207.

Buster — the Robot

Several recent issues of the Amateur Computer Group of New Jersey's newsletter carried details for construction and programming of a motor-driven "robot." Based on a KIM-1 microprocessor system, the robot can be best described as three-wheeled triangular box which can be directed with great precision by either a joy-stick or a program stored in memory. The microprocessor senses the output of a potentiometer-comparator feedback circuit to make the necessary angular correction to the steering system and determines control voltages to be applied to the wheel motors. Who built this innovation — a 14 year old named Tod Loofbourrow (with some help from a TAB Books publication describing a similar robot using discrete logic). Write: Amateur Computer Group of New Jersey, UCTI, 1776 Raritan Road, Scotch Plains, NJ 07076 for details if this type of project interests you.

New Products

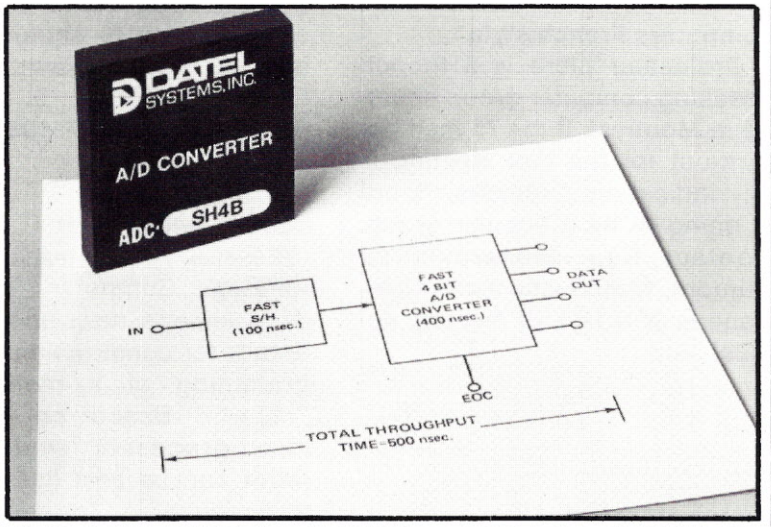


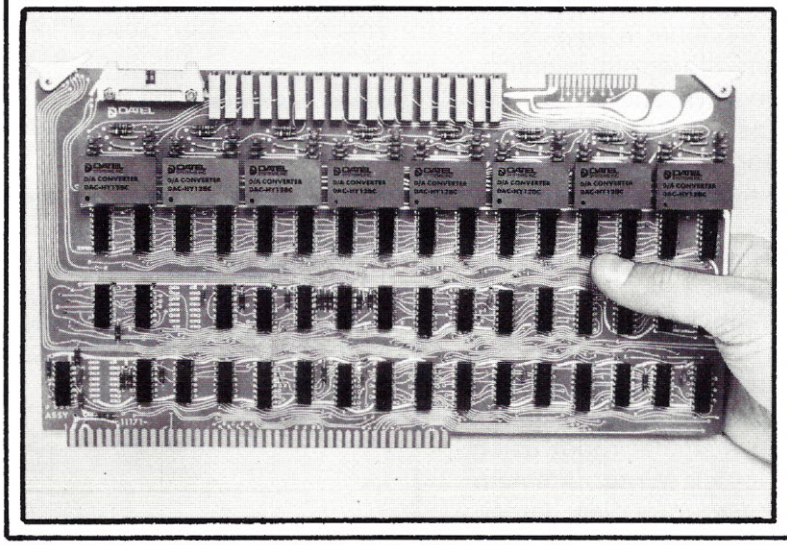
low cost CMOS A/D converter

Designed for applications requiring low power, accurate A/D conversion from slowly varying inputs, the ADC-ED series from Datel is priced from $13.50 for a single 8 - bit A/D with 0.2% relative accuracy. Conversion time is 1.8 msec. The EK converters draw 20 mW from a ±5 VDC supply. Packaged in standard 24 - pin ceramic DIP packages, the converter series consists of three binary - coded output models with 8, 10, and 12 bit resolutions. Datel Systems, Inc., 1020 Turnpike St., Canton, Mass. 02021.

ultra - fast A/D converter has built - in sample - hold

Priced at $79 in single quantities, Datel's model ADC-SH4B 4 - bit successive approximation converter contains a 100 nanosecond acquisition sample - hold circuit. A/D conversion time is 400 nanoseconds. Nonlinearity is ±2% maximum. The analog input range is 0 to +1 volt. Supply requirement is ±15 VDC and + 5 VDC. The unit is packaged in a 2 X 2 X 0.375 inch module. Contact Datel at 1020 Turnpike St., Canton, Mass. 02021.



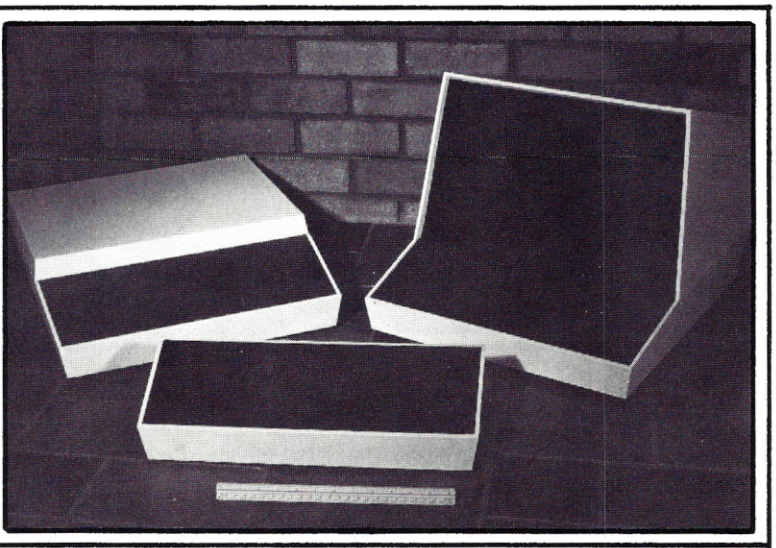


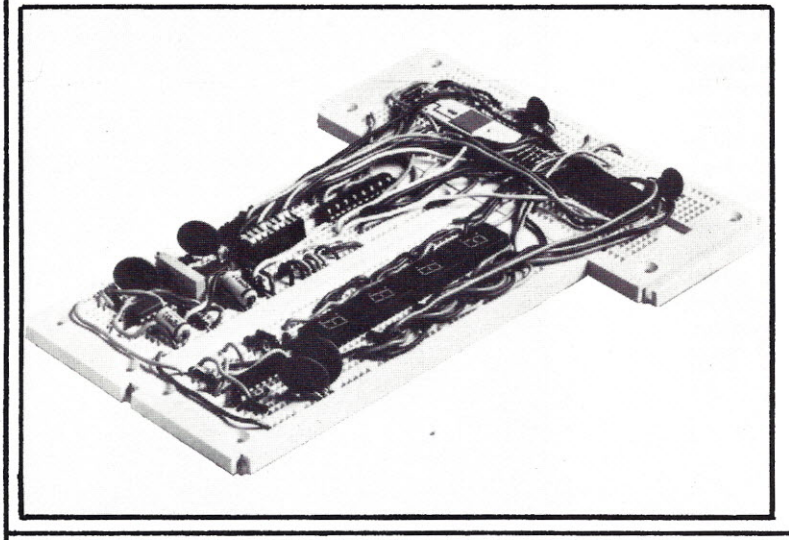
8 channel D/A card for Intel SBC-80/10 or MDS-800 bus

Industrial process control microcomputers often require one D/A output for every A/D input channel to prmit computer assisted feedback control. Datel ST-800-DA8 contains 8 D/A channels on a stand-alone 12" X 6.75" X 0.375" card. Settling time of each 12-bit D/A is 4 microseconds. Two or more 2-byte instructions can be used to update each D/A channel. Included is a jumper programmable interrupt. When the last channel is reached, the processor is interrupted. Available for $695 from Datel, 1020 Turnpike St., Canton, Mass. 02021.

inexpensive molded enclosures for CRTs and keyboards

Model VTE 101 is a CRT enclosure which measures 19 X 21 X 4 inches at the base. The CRT shroud with smoke gray plexiglass screen measures 11.25" high, 17" wide, and 12.5" deep. Model TVTE 101 consists of the same base as the CRT enclosure with a cover. Suitable for a keyboard TV typewriter. Model KBE 101 will house just the keyboard. Priced at $77.75, $44.45, and $22.25 respectively. Write to: Enclosure Dynamics, Inc., P.O. Box 6276, Bridgewater, N.J. 08807.



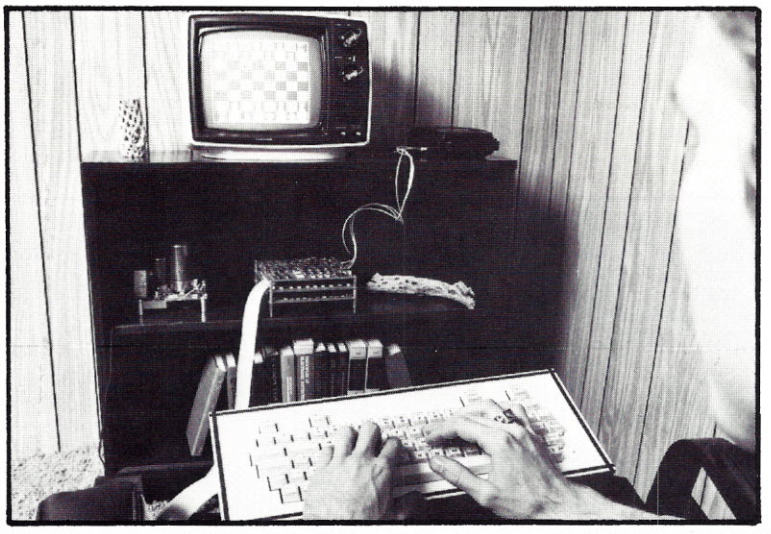


solderless breadboards for SSI/MSI and LSI DIP packages

New interlocking solderless breadboards from Continental Specialties permit optimum layout configurations for added economy. Each 6" Experimentor breadboard contains 94 5-point contacts. Experimentor model 600 has a 0.6" wide channel to seat LSI chips. Experimentor 300 seats standard DIPs and is priced at $9.95. Contact Continental Specialties at 44 Kendall St., Box 1942, New Haven, CT. 06509 (203) 624-3103.

complete direct - etch and photo - resist PC board kits

Direct - etch boards are made by applying the acid resistant circuit patterns on the bare copper boards and etching with the ferric chloride in the kit. To make duplicate boards, layout the patterns on the mylar sheet. Expose the photosensitive boards supplied to the sun while covered with the mylar art master, and etch. Kit 32X - 1 makes two boards for $11.50. Seven boards can be made with kit 32Xa-1 for $28. Vector Electronics Co., 12460 Gladstone Avenue, Sylmar, Ca. 91342.

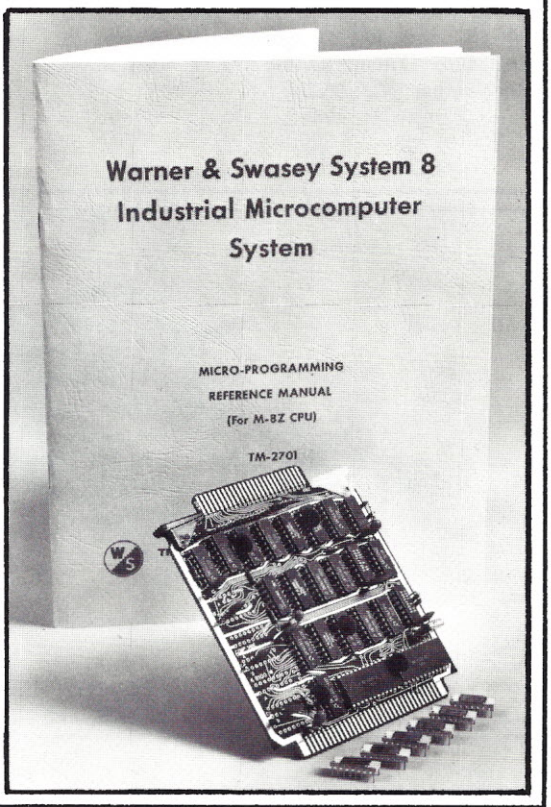


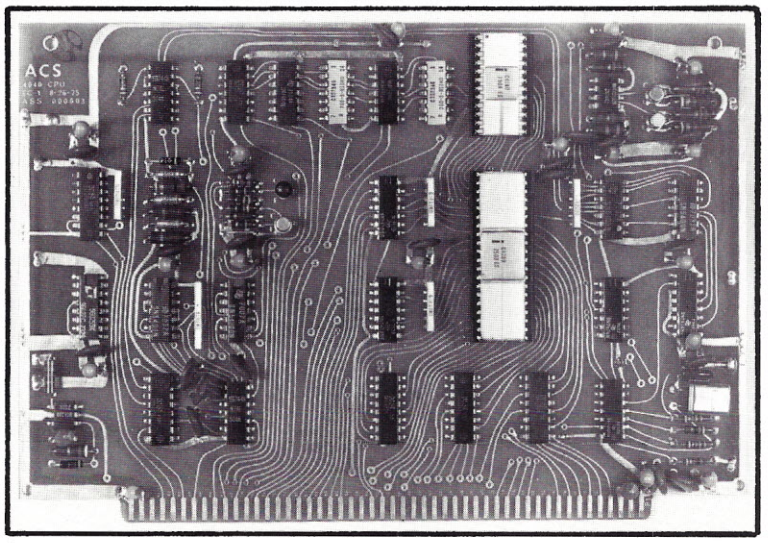
microprogrammable CPU for critical applications

Designs requiring a custom instruction set or special purpose algorithms for high speed execution are a few of the tasks which can be handled by the M - 8Z micro - programmable CPU. The M - 8Z has a micro - cycle time of 250 nanoseconds and interfaces with all modules in the System 8 line from Warner & Swasey. The module is priced at $285, including a set of unprogrammed instruction chips for custom designs, a micro assembler (FORTRAN), and a microprogramming manual. Write: The Warner & Swasey Co., Computer Division, 7413 Washington Ave. So., Edina, Minn. 55435.

microsystem offers user defined keyboard and extensive software

Designed for maximum operating flexibility, MICROMIND's 80 key keyboard is functionally defined with software to fit the user's application. Many types of characters can be keyed: standard 5 X 7 or 7 X 9 ASCII, APL, Japanese, Greek, Hebrew, chess pieces, etc. Custom keyboard arrangements can be configured with the relegendable keycaps supplied with the system. Software includes an interactive editor, assembler, monitor, cassette-based file system, and an extended form of BASIC language. Also, games and utilities are supplied. MICROMIND can be used as an intelligent terminal, remote job entry station, graphics display system, etc. Complete with a 6500A microcomputer, character generator, 8K of memory, input/output interface board, power supply, and 80 key keyboard, the only additional items needed are a standard television set and an inexpensive cassette recorder. Priced at $987.54, MICROMIND is assembled and tested. Write ECD Corp., 196 Broadway, Cambridge, Mass. 02139.



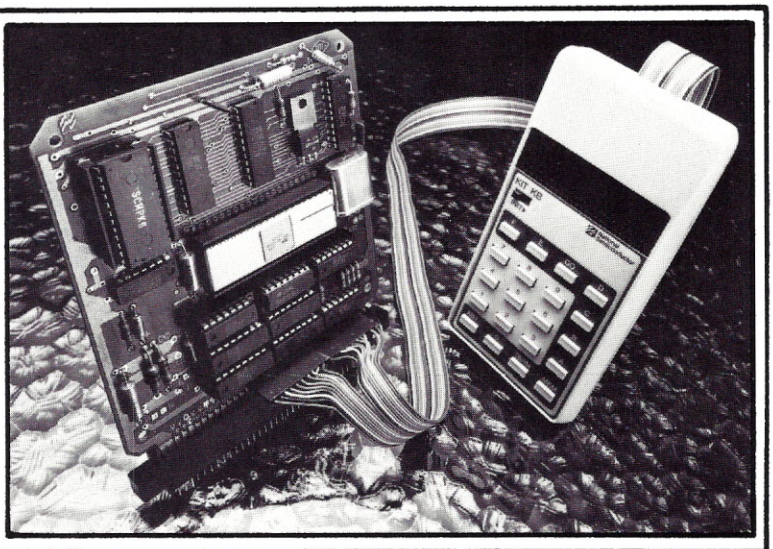


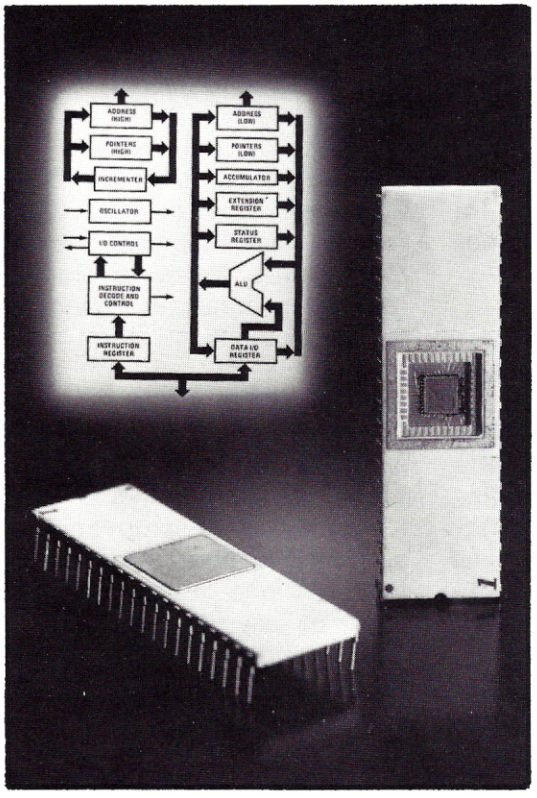
4 - bit general purpose microcomputer

The ACS - 4040 microcomputer is a parallel 4 - bit system suitable for process control, data acquisition, and front end communications controllers. Included on the ACS - 4040 board is a crystal clock, 8 bank RAM addressing, 2 bank ROM addressing, buffered expansion bus, I/O bidirectional expansion bus, Run/Single-Step mode control, Single Step input, photoisolated local/remote power and master reset. Requires +5 VDC, 0.9 Amps and -10 VDC, 0.1 Amp. Priced at $250 (1 - 9 units): Automated Computer Systems, 2361 E. Foothill Blvd., Pasadena, Ca. 91107.

SC/MP-II: fast, low power, single voltage processor

Samples are now available of a new N - channel MOS version of the "SC/MP" 8 - bit single chip microprocessor. The new SC/MP requires less than 200 milliwatts from a single +5 VDC source. Typical instruction execution time is 5 microseconds - one microsecond is required to complete one microcycle. The SC/MP - II is said to be fully compatible with its predecessor in terms of pin configuration, object code, and software. Write: National Semiconductor, 2900 Semiconductor Drive, Santa Clara, Ca. 95051.



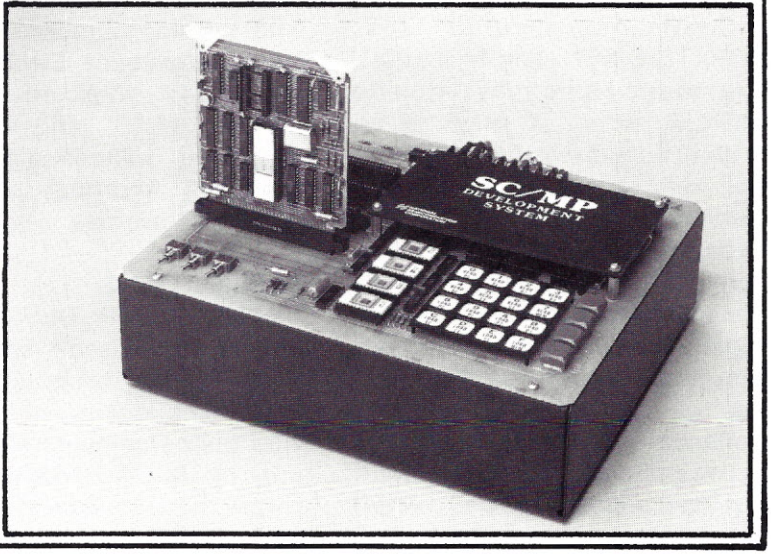


SC/MP hand - held terminal kit for $95

The SC/MP Keyboard Kit is useful for engineers and hobbyists who wish to program the SC/MP microprocessor kit without the added expense of a teletype system. The heart of the SC/MP Keyboard Kit is a ROM firmware package of 512 bytes which replaces the "Kit Bug" ROM supplied with the SC/MP microcomputer kit. The new ROM allows the effective use of the hexadecimal keyboard, to execute programs, to examine or modify the contents of memory and the SC/MP registers, and to monitor program performance. The terminal has twenty functional keys: 16 keys for hex values 0 through F, Abort Command, Memory Command (read next memory location), Go Command, and Terminate Command (change mode of current operation). Complete with all required integrated circuits, resistors, keyboard display cable connector assembly, wire wrap connectors and wire wrap tool. National Semiconductor, 2900 Semiconductor Drive, Santa Clara, Ca. 95051.

SC/MP low - cost development system

Hardware and software designs based on the SC/MP 8 - bit microprocessor can be developed and tested with the "Low-Cost Development System" (LCDS) from National Semiconductor. ROM - based firmware on the motherboard allows the user to alter the SC/MP registers and memory locations, run programs, or execute single instructions. The firmware subroutines allow entry of software debug commands via the control and display panel or an optional Teletype. $499 from: National Semiconductor, 2900 Semiconductor Drive, Santa Clara, Ca. 95051.



New Software

MICRO BASIC PLUS

Technical Systems Consultants, Box 2574 W. Lafayette, In. 47906 has developed a highly compact version of BASIC which offers a wide range of statements, commands, and functions. Designed to run in a minimum of 4K of memory (which includes up to 60 user statements), MICRO BASIC PLUS executes much faster than most compact BASIC language packages by avoiding the slow 2-level interpretation technique commonly used to implement BASIC.

STATEMENTS

Statements available to the user include PRINT, INPUT, READ, DATA, RESTOR, IF. . .THEN, GOTO, GOSUB, RETURN, ON. . .GOTO, ON. . .GOSUB, FOR, with positive and negative STEP, NEXT, LET, DIM, for both single and double dimensioned arrays (up to 98 by 98), EXT, REM, and END.

COMMANDS & FUNCTIONS

Commands include LIST, SCRATCH, RUN, and MONITOR. Arithmetic functions include SIGN(X), where X may be an arithmetic expression; this function returns a value of +1 for positive arguments, -1 for negative arguments, and Ø if X is zero. The ABS function returns the absolute value of any argument. RND is a random number function: whenever RND appears in an expression it will be replaced by a random number between Ø and 99. TAB is an output formatting function: TAB(X) will move to column X, where X can be any arithmetic expression. Function SPC(X) can be used to output a specified number of spaces.

OPERATORS

Math operators include minus, plus, multiplication, division, exponentiation, addition, and subtraction. The arithmetic range is + 99999. Relational operators include equal, less than, greater than, less than or equal to, greater than or equal to, not equal.

FEATURES

A line editor is included which permits lines to be inserted, deleted, or added at any time. Lines may be entered in any sequence; the interpreter automatically puts them in ascending order. Line lengths are limited to 72 characters. If this is exceeded, the line entered is not stored and a new "enter line" prompt is printed. Multiple statements per line are permitted using a ":" as the separator. Error identification includes 18 error codes which are printed along with the line number of the defective statement. Errors detected include syntax errors in the INPUT, READ, and IF statements, as well as many other cases of improper syntax or arithmetic overflow.

APPLICATIONS

MICRO BASIC PLUS will run on any 6800 based system which contains the Motorola MIKBUG ROM. The documentation sold with the commented source listing of MICRO BASIC PLUS shows how to adapt this compact BASIC to run on systems which do not contain the MIKBUG routines. The commented source listing, available from Technical Systems Consultants (TSC); for $15.95, also includes a programming example, dumping and loading procedures using paper tape or cassette, and error codes. MICRO BASIC PLUS is also available on the "Kansas City Standard" cassette for $6.95 and paper tape for $6.00. Write for the new Technical Systems Consultants Software Catalog (25¢ for postage and handling) for more details.

MICROCHESS

An unusual software offering is MICROCHESS, designed to play the game of chess on a KIM-1 6502 microprocessor system. The program can be adjusted to one of three different levels of play requiring 3, 10, or 100 seconds for each computer move. Although the level of play is probably below that of the serious chess player, it is a good match for the average computer programmer.

MICROCHESS program and data occupy only 1100 of the 1152 bytes of available memory on the standard KIM-1 microcomputer. All moves are entered and results displayed via the KIM-1 keyboard and hex LED display. Complete with a Player's Manual, annotated source listing, program documentation describing the strategic algorithms, instruction for modifying or expanding the system, MICROCHESS is available for $10 from: MICROCHESS, 1612-43 Thorncliffe Pk Dr, Toronto, Ont., M4H 1J4, Canada.

Advanced Microcomputer Products Presents

CONTINENTAL SPECIALTIES CORPORATION

CSC Prototype Design Equipment

QT SOCKETS AND BUS STRIPS

Universal breadboarding system eliminates the soldering iron. QT sockets provide terminals, each consisting of five pre-connected tie-points, running across the width of the socket. QT bus strips consist of two rows of connected tie-points (in groups of five) which run the length of the bus strip. All QT sockets and bus strips are molded of high-temperature plastic so that leads and wires can be soldered while plugged in. Specially engineered, large-aperture holes guide leads into contacts easily and effortlessly. Contacts are pre-stressed, spring-loaded, non-corrosive, nickel silver alloy. Average contact resistance is 0.5 milliohm initially; 0.4 milliohm after use. Any diameter lead from .015" to .032" may be used. All QT units are .33" thick; all sockets are 1.25" wide; all bus strips are .36" wide. Units mount with 4-40 flat-head screws from the front or 6-32 self-tapping screws from the rear. QT units feature unique snap/lock mechanism which mates units together in seconds to expand or contract a breadboard system at will.

CSC No.	Length, Ins.	Terminals*	Mtg. Dimens., Ins.	List Price
QT SOCKETS				
QT-59S	6.5	118	6.2 x 0.7	$12.50
QT-47S	5.3	94	5.0 x 0.7	10.00
QT-35S	4.1	70	3.8 x 0.7	[illegible]
QT-18S	2.4	36	2.1 x 0.7	4.75
QT-12S	1.8	24	1.5 x 0.7	3.75
QT-8S	1.4	16	1.1 x 0.7	[illegible]
QT-7S	1.3	14	1.0 x 0.7	[illegible]
QT BUS STRIPS				
QT-59B	6.5	20	6.2 x 0.7	$ 2.50
QT-47B	5.3	16	5.0 x 0.7	[illegible]
QT-35B	4.1	12	3.8 x 0.7	2.00

*Groups of five pre-connected tie-points.

PROTO-BOARD® UNITS

All the speed and convenience of QT Sockets and Bus Strips in both kits and preassembled units. Assemble, test and modify circuits literally as fast as you can think.

PROTO-BOARD 6 KIT
Contains one preassembled QT-47S socket, two preassembled QT-47B bus strips, four 5-way binding posts, metal ground/base plate, non-marring feet and all hardware. Ten minute assembly time. Size: 6" l. x 4" w. x 1.4" h. Weight: 7 ozs.
CSC Model PB-6 Kit—Complete. List Price $15.95

Now in stock for
Microprocessor Prototyping
NEW SOCKETS & BUS STRIPS
Experimenter 600 . . . . . $10.95
.6 spacing for microprocessor applications
Experimentor 300 . . . . . $ 9.95
.3 spacing for microprocessor applications

PROTO-BOARD 100 KIT
Contains two preassembled QT-35S sockets, one preassembled QT-35B bus strip, two 5-way binding posts, non-metallic base plate, non-marring feet and all hardware. Ten minute assembly time. Size: 4.5" w. x 6" l. x 1.4" h. Weight: 7.5 ozs.
CSC Model PB-100 Kit—Complete. List Price $19.95

PROTO-BOARD 101
Fully assembled breadboard contains two QT-35S sockets and four QT-35B bus strips mounted on metal ground/base plate with non-marring feet. Excellent for audio and small digital projects. Size: 5.8" l. x 4.5" w. x 1.4" h. Weight: 9 ozs.
CSC Model PB-101 Breadboard—Complete. List Price $29.95

PROTO-BOARD 102
Fully assembled breadboard contains two QT-47S sockets, three QT-47B bus strips and one QT-35B bus strip on a metal ground/base plate with non-marring feet. Excellent for intermediate digital needs. Size: 7" l. x 4.5" w. x 1.4" h. Weight: 10 ozs.
CSC Model PB-102 Breadboard—Complete. List Price $39.95

PROTO-BOARD 103
Fully assembled breadboard contains three QT-59S sockets, four QT-59B and one QT-47B bus strips, four 5-way binding posts on a metal ground/base plate with non-marring feet. Build calculators, interfaces, networks, etc. Size: 9" l. x 6" w. x 1.4" h. Weight: 1¾ lbs.
CSC Model PB-103 Breadboard—Complete. List Price $59.95

PROTO-BOARD 104
Fully assembled breadboard contains four QT-59S sockets, seven QT-59B bus strips and four 5-way binding posts on a metal ground/base plate with non-marring feet. Build a CPU, encoder, complex display, etc. Size: 9.8" l. x 8" w. x 1.4" h. Weight: 1¾ lbs.
CSC Model PB-104 Breadboard—Complete. List Price $79.95

PROTO-BOARD 203
Fully assembled breadboard contains built-in, short-proof, fused, 5 VDC at 1 amp, regulated power supply, in addition to three QT-59S sockets, four QT-59B bus strips, one QT-47B bus strip and four 5-way binding posts. Capacity for most digital and many analog projects. Size: 9.75" l. x 6.6" w. x 3.25" h. Weight: 5 lbs.
CSC Model PB-203 Breadboard—Complete. List Price $75.00

PROTO-BOARD 203A
Provides all the features of Proto-Board 203 with additional +15 and −15 VDC at 0.5 amp power supplies with internally adjustable output voltages. Size: Same as PB-203. Weight: 5.5 lbs.
CSC Model PB-203A Breadboard—Complete.
List Price $120.00

LOGIC PROBE LP-1

A compact, versatile design and test tool for digital applications. By simply connecting the clip leads to the circuit power supply, setting the logic family switch to the proper position (TTL/DTL or CMOS) and touching the probe tip to the IC pin in question, the LP-1 will give an instant picture of circuit conditions. Two level detector LED's (HI for logic "1" and LO for logic "0") and a blinking PULSE detector LED display the signal activity. One-shot, low-repetition-rate, narrow pulses are stretched to ⅓ second. A PULSE/MEMORY switch allows one shot events positive or negative transitions to be stored indefinitely. High frequency signals cause the PULSE LED to blink on and off at a 3 Hz rate. When the high frequency signals are unsymmetrical, there is an additional indication. Duty cycles less than 30% activate the LO LED; duty cycles above 70% activate the HI LED. Input Impedance: 100,000 ohms. Threshold: DTL/TTL, 2.25 V ±0.15 V equals "Logic 1", 0.80 V ±0.01 V equals "Logic 0"; HTL/CMOS, 70% of Vcc equals "Logic 1", 30% of Vcc equals "Logic 0". Minimum Detectable Pulse Width: 50 nanoseconds. Input Frequency: Up to 10 MHz. Input Overload Protection: ±50 V continuous; 117 VAC for less than 15 seconds. Size: 5.8" l. x 1.0" w. x 0.7" d. Weight: 3 ozs. Power Required: 30 mA at 5 VDC; 40 mA at 15 VDC; 36 V max. with polarity reversal protection.
CSC Model LP-1 Logic Probe—Complete. List Price .. $44.95

WE ARE A FULL LINE CSC DISTRIBUTOR. WE CAN SUPPLY ANY PRODUCT OUT OF THE CSC CATALOGUE. IF YOU NEED THEIR LATEST CATALOGUE WRITE AND WE'LL SEND IT ALONG.

8K RAM BOARD $198.00

ADVANCED ★MICROCOMPUTER PRODUCTS

BUILD YOUR OWN COMPLETE COMPUTER SYSTEM

★ AMI EVK 99 6800 BASED MICROCOMPUTER

Ask the Home Brew Computer Club in Santa Clara, California why they choose the EVK 99 Microcomputer. They now have 85 systems. The reasons are clear: Performance, versatility and cost effectiveness. Photo is of expanded EVK 99 Board.

Photo of expanded EVK 99

$159.95

1. EVK 99 consists of the following only
   - Qty. (1) PCB 10½ x 12
   - (1) 6800 8 bit CPU
   - (1) 6850 ACIA (Asynchronous Communication Interface Adapter)
   - (1) 6820 PIA (Peripheral Interface Adapter)
   - (2) 6830 ROM Subroutine Program Library (1) 6831 16K ROM
   - (4) 6810-1 RAM
   - (1) Operating Manual & Instructions

   To implement fully expanded system additional IC's are required. (See Photo)
2. UNIVERSAL KLUGE BOARD $98.00
   Completely compatible to the EVK system. This board allows the user to add any additional parts he desires. It has 5400 holes, holds 94 16 pin IC's or others on .2, .4, .6, .9" or greater centers. 2 · 50 pin, 1 · 20 pin flat cable, 1 · 25 pin RS232 connector.
3. 16K BYTE RAM BOARD $77.95
   16K x 8 or 8K x 16 organization. Uses 2102's or 21L02's and adapts to slow memory.
4. 6 SLOT MOTHER BOARD $29.95
   Connectors mount on .75" centers. May be plugged together for expanded applications.
5. EXTENDER BOARD $46.50
   For use with back plane above in card cage configuration.

New boards are now in development including cassette I/O and others. If you want more information drop us a line.

★ MICROPROCESSOR PRODUCTS

8080A/9080A	$24.95	S6860			$15.95	21L02/91L02	1.99
AMI S6800	24.95	S6820			11.00	2102/9102	1.89
AM2901	29.95	S6810-1			6.25	9111/2111	4.75
GI CP1600	69.95	S6850			10.95	9101/2101	5.25
Z80	69.95	S2350			12.95	9112/2112	2.95
6502	29.95	S1883			5.90	7489	1.88
AM2902PC	$ 3.95	S6834-1			22.95	4104/4804	12.95
AM2905PC	10.95	S6800 Programming Manual			15.00	4402A	12.95
AM2906PC	11.95	S6800 Hardware Manual			15.00	4200	12.95
AM2907PC	10.75	Tiny Basic (paper tape)			22.95	9131APC	14.95
AM2909PC	24.95	On 6834-1 Prom			175.00	9141APC	14.95
AM2911PC	18.95	8212	$ 4.75	AM2533	7.50	9050/TMS4050	12.95
AM2918	6.75	8224	6.25	AM2401	16.50	9060/TMS4060	12.95
AM29720PC	7.95	8216	4.95	AM1402	6.95	P3101	3.25
AM29721PC	7.95	8226	5.95	AM1403	5.95	5101	11.95
AY-51013A	5.50	8228	9.90	AM1404	6.95	1702A	10.90
AM2812	14.95	9551	19.95	MM500H	2.25	2708	68.00*
AM2841	12.00	9555	19.95	MM5016H	4.95	5204	23.50
AM3341	9.95			AM3347	6.40		

TV GAME KITS

ONLY $29.95
Qty (1) GI AY38500-1 TV Game Chip

TV KIT NO.1 — ONLY $34.95
Includes AY38500-1 TV GameChip, PC Board for video composite output plus instructions. Add $5.95 for 2MHz crystal.

TV KIT NO. 2 — ONLY $49.95
Includes AY38500-1 TV GameChip, PC Board, Switches, pots, speaker, crystal, XFMR, Bridge and all parts required for video composite. Works super with terminal. Includes instructions and data sheets.

SPECIAL
BUY 10 OF THE FOLLOWING AND YOUR PRICE IS. . .
AMD 8080A.............$19.95
AMD 9111.....................4.10
AMD 9112.....................2.30

RF MODULATOR MODULE

To convert your video composite to RF try
UM 1021 Channel 3/4 VHF Modulator $8.95
● DC Switchable ● DC Coupled modulation input with
VSWR = 2 max positive transfer characteristic.
VCC = 6.5 volts Designed for use with TV Games.

SPECIALS

75452	10/2.50
8T26B	2/5.00
74LS138	1.49
8097	.99
74166	.80
74161	.49
741HC	10/2.50
MH0026CN	3.95
1488	1.29
1489	1.29

★ NEW PRODUCTS

NEW FROM NEC Microcomputer! FIRST TIME OFFERED

D372 Floppy Disc Controller Chip	$125.00
D371 Mag. Tape Cassette Controller Chip	99.95
Application Notes (26 pages)	6.95
GI AY3550 4½ Digit DVM Chip	29.95
GI AY3600 Keyboard Encoder	9.95
Intersil 8038CC	4.95
XR2263 Servo Cont.	6.95
RO 32513 Char Gen	9.95

★ IC MARKET PLACE ★

★ LINEAR

LM105	3.50	LM320K-5.2	1.45	LM381N	1.79
LM106H	2.50	LM320K-12	1.45	LM382N	1.79
LM300H	.80	LM320K-15	1.45	LM555CN	.58
LM301H	.45	LM320T-5	1.75	LM556CN	1.30
LM301CN	.45	LM320T-5.2	1.75	LM567CN	1.45
LM302H	1.25	LM320T-8	1.75	LM565N	1.95
LM304H	1.20	LM320T-12	1.75	LM565H	1.95
LM305H	.85	LM320T-18	1.75	LM703CN	.45
LM305AH	1.25	LM232K-5	9.95	LM709H	.40
LM306N	.95	LM324M	1.65	LM709N	.40
LM307CN	.35	LM339N	1.55	LM710N	.70
LM307H	.35	LM340K-5	1.95	LM711N	.39
LM308H	1.00	LM340K-8	1.95	LM711H	.39
LM308CN	1.00	LM340K-12	1.95	LM723N	.55
LM309H	1.10	LM340K-15	1.95	LM723H	.55
LM309K	1.25	LM340K-18	1.95	LM725H	3.25
LM310CN	1.15	LM340K-24	1.95	LM733N	1.00
LM310H	1.15	LM351CN	.65	LM733CH	1.45
LM311H	.90	LM370N	1.65	LM739N	1.29
LM311CN	.49	LM370H	1.65	LM741CH	.35
LM312H	1.95	LM373N	3.25	LM741CN-14	.39
LM318H	1.50	LM377N	4.00	LM741CD	.49
LM318CN	1.50	LM380N	1.39	LM747H	.75
LM319N	1.30	LM380CN	1.05	LM747N	.75
LM320K-5	1.45			LM748H	.39
				LM748N	.39

LM1414N	1.75	75324	3.50
LM1458CN	.65	SG4501T	2.40
LM1496N	.95	SG4501N	2.40
LM2111N	1.95	SG3524	8.95
LM2901N	1.95	RC4194T	6.95
LM3065N	.69	RC4195T	6.95
LM3900N	.55	RC1458CN	.89
LM3909	1.10	RC1556CN	.99
LM7520	2.25	RC3403AD	2.30
LM7524N	1.75	RC4131CN	.99
LM7525N	1.90	RC4136D	1.90
LM7534N	2.20	RC4558CN	.99
LM7535N	1.25	RC4151CN	6.50

★ 7400 TTL

7400N	.18	7445N	.99	74122N	.44	74185N	$2.10
7401N	.18	7446N	.99	74123N	.60	74190N	$1.30
7402N	.18	7447N	.99	74125N	.79	74191N	$1.30
7403N	.18	7448N	.99	74126N	.79	74192N	$1.10
7404N	.19	7450N	.26	74132N	.70	74193N	$1.10
7405N	.20	7451N	.26	74141N	.88	74194N	.95
7406N	.25	7453N	.26	74145N	.70	74195N	.95
7407N	.25	7454N	.26	74147N	$1.59	74196N	.95
7408N	.21	7470N	.45	74148N	$1.29	74197N	.95
7409N	.21	7472N	.30	74150N	$1.15	74198N	$1.45
7410N	.21	7473N	.30	74151N	.98	74199N	$1.85
7411N	.21	7474N	.30	74153N	.98	74251N	$1.45
7412N	.21	7475N	.69	74154N	$1.15	74279N	$1.10
7413N	.45	7476N	.42	74155N	.95	8093N	.49
7414N	.69	7480N	.70	74156N	.95	8094N	.49
7416N	.35	7482N	.95	74157N	.95	8095N	.99
7417N	.35	7483N	.95	74160N	.99	8096N	.99
7420N	.19	7485N	.89	74161N	.99	8097N	.99
7421N	.35	7486N	.28	74162N	.99	8098N	.99
7423N	.35	7489N	$1.88	74163N	.99	75150N	$1.70
7425N	.35	7490N	.60	74164N	.99	75450N	$1.10
7426N	.35	7491N	.70	74165N	.99	75451N	.59
7427N	.35	7492N	.59	74166N	$1.10	75452N	.59
7428N	.28	7493N	.79	74170N	$4.20	75453N	.59
7430N	.21	7494N	.79	74172N	$16.25	75454N	.59
7432N	.31	7495N	.79	74173N	$1.40	75491N	.70
7433N	.30	7496N	.79	74174N	$1.25	75492N	.70
7437N	.35	7497N	$3.75	74175N	$1.25	75493N	$1.10
7438N	.35	74100N	$1.10	74176N	.79	75494N	$1.10
7440N	.21	74104N	.53	74177N	.79		
7441N	$1.06	74105N	.53	74180N	.79		
7442N	.69	74107N	.33	74181N	$2.75		
7443N	.69	74109N	.33	74182N	.79		
7444N	$1.50	74121N	.35	74184N	$2.10		

100 PIECES MIX TAKE 10% DISCOUNT
1000 PIECES MIX TAKE 15% DISCOUNT

★ CMOS

CD4000	.25	CD4026	1.95	CD4075	.35
CD4001	.25	CD4027	.85	CD4078	.35
CD4002	.25	CD4028	1.50	CD4081	.35
CD4006	1.75	CD4029	1.99	CD4082	.35
CD4007	.25	CD4030	.60	CD4510	1.95
CD4008	1.95	CD4033	2.49	CD4511	1.40
CD4009	.48	CD4034	4.95	CD4516	1.95
CD4010	.48	CD4035	1.79	CD4518	2.45
CD4011	.28	CD4040	2.10	CD4520	2.49
CD4012	.28	CD4042	1.45	CD4522	2.56
CD4013	.60	CD4043	1.45	CD4526	2.56
CD4014	1.95	CD4044	1.95	CD4527	2.70
CD4015	1.95	CD4049	.61	CD4528	1.50
CD4016	.59	CD4050	.61	CD4531	1.99
CD4017	1.25	CD4051	2.25	CD4543	2.25
CD4018	2.39	CD4066	1.20	CD4555	1.75
CD4019	1.95	CD4068	.35	CD4556	1.75
CD4020	1.35	CD4069	.35	CD4581	4.95
CD4021	1.85	CD4070	.99	CD4582	1.95
CD4023	.29	CD4071	.35	CD4585	2.45
CD4024	1.29	CD4072	.35	Also	
CD4025	.29	CD4073	.35	5101	$11.95

PLESSEY SEMICONDUCTORS

RADIO COMMUNICATION

SL610C	1.95	SL624C	2.75
SL611C	1.95	SL630C	1.95
SL612C	1.95	SL640C	1.95
SL613C	2.40	SL641C	6.95
SL620C	2.55	SL645C	5.95
SL621C	2.55	SL650C	5.75
SL622C	2.95	SL651C	4.95
SL623C	2.95	SL680C	3.75

Application Manual for SL600 Circuits 1.95
Plessey Semiconductor Catalogue 3.95

DIVIDERS

SP8600B	6.95	SP8620B	12.50	SP8640B	6.20
SP8601B	4.25	SP8621B	8.50	PS8641R	7.50
SP8602B	16.00	SP8622B	6.50	SP8642B	12.50
SP8603B	14.50	SP8630B	20.00	SP8643B	20.00
SP8604B	8.25	SP8631B	12.00	SP8646B	6.50
SP8607B	17.00	SP8632B	10.00	SP8647B	8.00
SP8613B	17.50	SP8634B	35.00	SP8650B	25.00
SP8614B	18.00	SP8635B	27.50	SP8651B	18.50
SP8615B	20.00	SP8638B	22.50	SP8652B	12.00
SP8616B	22.00	SP8637B	18.00		

ALTAIR /IMSAI COMPATIBLE 8K MEMORY

Our new brain board comes complete with 21L02-1 memory chips (500ns) and sockets.

All address control and data out lines fully buffered.

THE SAME 8K BOARD OFFERED TO THE SCCS GROUP PURCHASE. NOW OFFERED TO YOU AT THE SAME LOW LOW PRICE!

$198.00 COMPLETE

If you desire 91L02APC - Please add $9.95

TMS 4060 RAM	$4.95
NEC 4096 RAM	4.25
1602B UART	5.50
1502F FIFO	12.95
34702 BAUD GEN	14.95
NEC 5101C—E CMOS RAM	6.95
RO32513 CHAR GEN Lower Case	14.95
WD 1671B ASTROS	29.95
WD 1941 DUAL BAUD GEN	9.95
WD 1771 FLOPPY DISC CONT.	74.95

SPECIAL

100 22PIN 4096 DYN RAM	3.25
100-22 PIN 4096 DYN RAM	3.25
100-16 PIN 4096 DYN RAM	3.45

6800 STARTER SET

Only 68 Bucks $68.00
You Get (1) 6800 8 bit CPU
(1) 6820 PIA
(1) 6850 ACIA
(16) 2102 RAM's

Add $15.00 for AMI 6800 Programming Manual

8080A STARTER SET

Only $80.00
You Get (1) 8080A 8 bit CPU
(1) 8212 I/O Port
(1) 8224 Clock
(1) 8228 System Controller
(24) 2102 RAM's

Add $5.95 for MCS 8 8080 Users Manual

FCN GEN KIT $18.50

63 Key Keyboard 19.95
19 Key Keyboard 11.95

Also . . . we stock entire continental specialties line new sockets and bus strips for microprocessor design

Experimenter 600 (.6 spacing) $10.95
Experimenter 300 (.3 spacing) 9.95

All shipments first class or UPS in U.S. or add 5% for shipping and handling. Minimum order $10.00 California Residents add 6% tax. Foreign add 8%.

ADVANCED ★MICROCOMPUTER PRODUCTS
P. O. BOX 17329
IRVINE, CALIFORNIA 92713
PHONE (714) 968-3655

★ ORDER YOUR NEW CATALOG NOW!

Z-80 $69.95 4K STATIC RAM★TV GAMES

TRANSISTORS

P/N	DESCRIPTION	PKG.	UNIT PRICE	10	100	1000
2N918	NPN High Frequency	TO-18	.24	.17	.16	.15
2N930	NPN Low Level Low Noise	TO-18	.24	.17	.16	.15
2N2222A	NPN General Purpose	TO-92	.19	.165	.155	.135
2N2219A	NPN General Purpose	TO-5	.23	.17	.16	.15
2N2369A	NPN High Speed Switch	TO-92	.23	.165	.155	.135
2N2905A	PNP Small Sig. General Purpose	TO-5	.22	.20	.18	.17
2N2907A	PNP General Purpose	TO-18	.21	.175	.165	.155
2N3392	NPN Small Signal G/P.	TO-92	.19	.15	.14	.13
2N3414	NPN Gen. Purpose Switch	TO-92	.20	.19	.185	.18
2N3563	NPN Small Signal HF	TO-92	.24	.165	.14	.13
2N3565	NPN Low Level High Gain	TO-92	.24	.165	.14	.13
2N3638	PNP Sm. Sig. GP Amp	TO-106	.24	.165	.14	.13
2N3640	PNP Hi Speed Sat. Switch	TO-92	.24	.165	.14	.13
2N3646	NPN Hi Speed Sat. Switch	TO-92	.24	.165	.14	.13
2N3904	NPN Sm. Sig. GP Amp.	TO-92	.23	.15	.14	.13
2N3906	PNP Sm Sig. GP Amp	TO-92	.23	.15	.14	.13
2N4092	Nch. Fet. Vp 20 BV40	TO-18	.48	.43	.41	.40
2N4093	Nch. Fet. Vp 20 BV50	TO-18	.50	.45	.43	.42
2N4123	NPN Sm. Sig. GP Amp	TO-92	.23	.15	.14	.13
2N4125	PNP Sm. Sig. GP Amp	TO-92	.23	.15	.14	.13
2N4400	NPN GP Switch	TO-92	.23	.155	.145	.135
MPSA13	NPN Darlington (hfe 5000)	TO-92	.28	.27	.24	.225
2N3055	NPN Power 15A 115 Watt	TO-3	.95	.90	.78	.72
MJ3055	Plastic (Fcn equiv. to MJE3055)	TO-220	.95	.90	.80	.77
2N3725	Core Driver	TO-5	.65	.50	.35	.78
FPQ3725CN	Quad Core Driver	DIP	2.50	1.95	1.75	1.40

BRIDGES

			1-99	100-499	500-999
W02M	.75A	200V	.74	.46	.35
W005M	.75A	50V	.70	.43	.34
W06M	.75A	600V	.82	.52	.46
W08M	.75A	800V	.88	.56	.47
W10M	.75A	1000V	1.05	.65	.49
KBPC25-02	25A	200V	2.50	2.25	1.95
KBPC25-04	25A	400V	2.60	2.30	1.99
KBPC25-06	25A	600V	2.75	2.35	2.10

$1.00 Minimum per line item.

★ CRYSTALS ★

Microprocessors Timebases TV Games Precision Crystals

Frequency	Case	Price
1.000 MHz	HC6/HC33	4.95
2.000 MHz	HC6/HC33	5.95
2.097152 MHz	HC6	8.95
4.000 MHz	HC18	4.95
5.000 MHz	HC18	4.95
6.000 MHz	HC18	4.95
10.000 MHz	HC18	4.95
13.000 MHz	HC18	4.95
14.31818 MHz	HC18	4.95
18.000 MHz	HC18	4.95
20.000 MHz	HC18	4.95
32.000 MHz	HC18	4.95
100Kc	HC13	12.95
3579.45 Kc	HC18	4.95

★ Exar

SPECIALISTS IN NEW TECHNOLOGY LINEAR CIRCUITS

★ FUNCTION GENERATOR KIT ★

The Function Generator Kit features sine, triangle and square wave; THD 0.5% typ.; AM/FM capability.

XR-2206KA Includes monolithic function generator IC, PC board, and assembly instruction manual 18.95

★ TIMING CIRCUITS ★

XR555CP	Monolithic Timer	.90
XR556CP	Dual 555 Timer	1.75
XR2556CP	Dual Timing Circuit	2.95
XR2240CP	Programmable Timer	4.20
XR320P	Timer	1.40

★ COMMUNICATION CIRCUITS ★

XR1488	Quad RS232 Driver	1.45
XR1489	Quad RS232 Receiver	1.45
XR2208	Four Quadrant Multiplier	4.75

★ PHASE-LOCKED LOOPS ★

XR2211CP	FSK Demod/Tone Decoder	5.95
XR567CP	Tone Decoder	1.80
XR25567CP	Tone Decoder	1.80

★ STEREO DECODERS ★

XR1310P	FM Stereo Demodulator	3.00
XR1310EP	FM Stereo Demodulator	3.00
XR1800	Stereo Demodulator	3.00

★ FUNCTION GENERATORS ★

XR2206CP	Monolithic Function Generator	5.25
XR2207CP	Current Controlled Oscillator	3.75

★ OPERATIONAL AMPLIFIERS ★

XR742CP	Zero Crossing AC Trigger	3.00
XR4558CP	Dual Hi Gain Op Amp	1.20
XR4136CP	Quad Op Amp	1.75
XR4202CP	Programmable Quad Op Amp	4.25

★ JUST OUT

XR2263 Servo controller for remote control applications . . $6.95

POWER SUPPLY: SPECIAL OFFER

+5 Volts 1.5A Regulated
+12 Volts 1.0A Unregulated
Special buy on OEM Type Power Supply
Comes complete with 7.5A 117V Power Transformer
All Mounted in Chassis.
Specification Available Upon Request

Only $14.95

★ MISCELLANY

RIBBON CABLE
8 Conductor

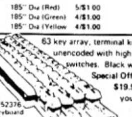
1-9 ft	10-49 ft	50ft up
.35/ft.	.25	.23

ELCO VARIPAK 2 H
2-Slot cardholders
4/$1.00

3.5V BUZZER $.95

LED'S
185" Dia (Red) 5/$1.00
185" Dia (Green) 4/$1.00
185" Dia (Yellow) 4/$1.00

KEYBOARD



63 key array, terminal keyboard, unencoded with high quality switches. Black w/white legends. Special Offer: Buy now at $19.95 and we will give you a free calculator type keyboard.

AY52376 Keyboard Encoder with keyboard Only $14.95

TOUCHTONE KEYBOARD FOR TELEPHONE

Use with MK5085 N integrated tone dialer & 3.579 Hz crystal. Crystal available for $4.95

Only $9.95

NOW!
A 19 key Number Keyboard compatible with our 63 Key Keyboard. Same manufacture and design. Keys for 0 to 9 and 110,300, FDX/HDX, minus Auto LF, Line Loc, Alt Char set and Char set lock.
ONLY $11.95 FIRST TIME OFFERED
(Identical to our 63 Key as pictured except 19 keys.)

★ CLOCK CHIPS

MM5311	24 Pin BCD Output 6 Digits	$4.50
MM5313	28 Pin 6 Digits	4.00
MM5314	24 Pin 6 Digits	3.50
MM5316	[illegible] 40 Pin Alarm Ledout	4.50
MM5375	Clock Chip	3.50

★ IC SOCKETS

Low Profile Tinplate

8 Pin	.17	24 Pin	.41
14 Pin	.20	28 Pin	.49
16 Pin	.22	40 Pin	.63
18 Pin	.31		

★ RESISTORS

Carbon Resistors 1/4 Watt ±5%
Most Standard Resistance Valves
NOTE Multiples of 5 pcs per valve minimum

Qty. (each valve)	Price
5-20	.05 each
25-50	.045 each
55-100	.04 each
105-500	.03 each
505 up	.025 each

★ VARIABLE RESISTORS

Centralab RN4 NAYSD503A (HMP 50K) $1.95
RN4 NAYSD503A (53C3 50K S) $1.95

★ CORCOM EMI FILTER

● CORCOM 2B4 2A Only $4.95
115/230V 50-400Hz Test 2100VDC
● CORNELL DUBILIER NF21015
2x4A 250 VAC Only $4.95
● DIP SWITCH
8 SPST Slide Switches in Dip $2.25

ROTARY SWITCH

Centralab PA 2001 $1.95

C&K V 25
Bat handle toggle switch
$1.50

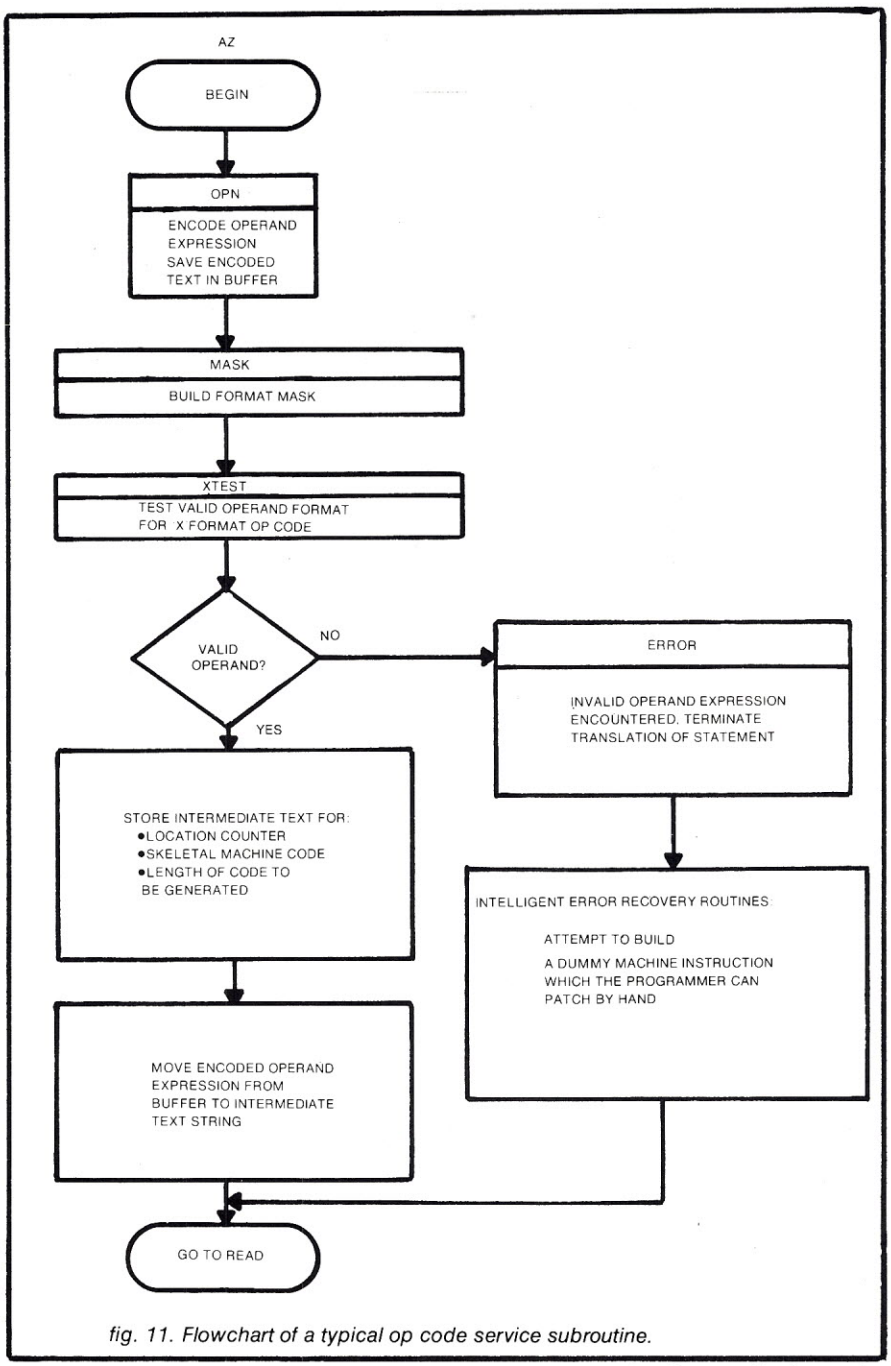


fig. 11. Flowchart of a typical op code service subroutine.

of translating this statement, the encoded operand expression, and two other important pieces of information.

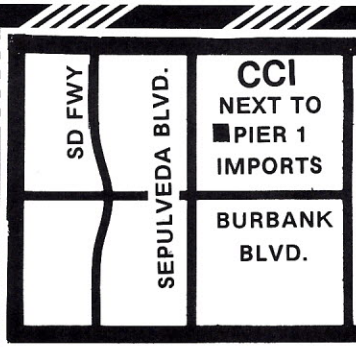
One of the pieces of information is a skeletal machine instruction for the op code. Typically, each instruction for a machine has a certain amount of fixed information which is dependent solely upon the op code and a certain amount of variable information which is dependent upon the operand expression, such as addressing modes used. One of the elements of intermediate text for a statement then is a skeletal form of the machine code to be generated with all its variable information set to zero.

The second piece of information is an operand pattern mask. It is, in fact, used to generate the first piece of information. Typically the operands for op codes in an assembly program must all conform to a general pattern. In the sample translator the operand can, at most, be of the format "D1(L1,X1),D2(L2,X2)". The "Dn" portions represent address or displacement expressions, the "Ln" portions are length expressions, and the "Xn" portions are index register expressions. Not all op codes can validly use this full form and most machine instructions will use only various subsets of this form, but the important thing is that all instruction operands must be in some subset of this syntactic form. The operand expression, "A+B/E(,3),X1" is of the form, "D1(,X1),D2" and the expression, "0(3),5(C2,25/3)" is of the form, D1(L1),D2(L2,X2)".

There is a subroutine, MASK, which is invoked after the operand expression has been encoded. This routine examines the encoded text and returns a (bit-wise) mask indicating which elements of the possible operand format were encountered in the operand. **Fig. 10** lists the bit definitions for the mask. By examining this pattern it can be immediately determined whether or not the operand has a valid syntax for the op code and which addressing options are used. This also indicates the form and length of the skeletal machine instruction to be produced. **Fig. 11** shows how a typical op code service subroutine functions. The operand is scanned and encoded, the syntax is checked, and the intermediate text for the statement is completed.

conclusion

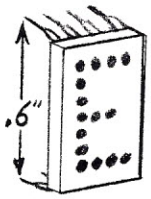
Label extraction and processing, and op code extraction and processing have been presented as a fairly simple and straightforward activity performed by a number of unique subroutines which we have examined at length. The next part of this series will deal with operand encoding which is the heart of phase one of the assembly language translator.



COMPUTER COMPONENTS

5848 SEPULVEDA BLVD.
VAN NUYS, CALIF. 91411
(213) 786-7411

HOURS
TUES. - FRI.
10 A.M to 9 P.M.
SAT. 9 to 9
SUN. 10 to 6
CLOSED
MONDAY



HEXIDECIMAL DISPLAY
HP 5082-7340

$7.00(1) $6.00(10) $5.00(100)

!! FACTORY PRIME INTEL 2708'S
1,024 BYTE EPROM
UV ERASEABLE
$65.(1) $50.(10) $40.(100)

74150 IN CERAMIC DIP HOUSE NUMBER
$1.00(1) 75¢(10) 60¢(100) 45¢(1000)

74181 IN CERAMIC DIP HOUSE NUMBER
$1.25(1) $1.00(10) 70¢(100) 50¢(1000)

74190 PRIME NATIONAL PARTS
90¢(1) 80¢(10) 70¢(100) 60¢(1000)

AMS 6002 CERAMIC DYNAMIC RAM 1Kx1
$1.00(1) 70¢(10) 50¢(100) 35¢(1K)

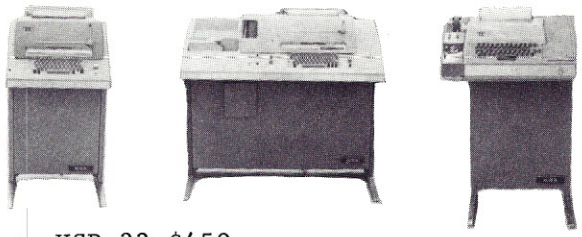
DB25's (S)$3.00 (P)$2.00 SHELLS$1.00
IMSAI 100 PIN CONNECTORS $5.00 ea.
DIP SW'S 8 POSN $2.50 7 POSN $2.25

20K SLIDE POTS (STACKPOLE 114)
50¢(1) 40¢(10) 35¢(100)

HPROM 1024 BY HARRIS SEMICONDUCTOR
256X4 FIELD PROGRAMMABLE ROM
THESE HAVE BEEN PROGRAMMED BUT MAY
HAVE HEAVY VIRGIN AREAS
25¢ EACH(1-1000) 15¢ EACH(1000 UP)

IMSAI

I8080-1K MICROPROCESSOR KITS
RAM 4A-4 MEMORY MODULE KITS
PIC 8 PRIORITY INTERRUPT & RTC KITS
SIO 2-1 SERIAL RS232 INTERFACE KITS
PIO 4-4 4 PARALLEL INPUT/OUTPUT PORTS
EXP 22 22 SLOT MOTHERBOARD



TELETYPES

* KSR 33 $450.
* KSR 35'S FROM $500. TO $1200.
* ASR 33'S FROM $600. TO $1000.
* ASR 35'S FROM $750. TO $2500.

"D" CELL
BATTERY HOLDER
65¢(1) 50¢(10) 40¢(100)

TO-3
HEAT SINK
50¢(1) 40¢(10) 35¢(100)

POLYMORPHIC SYSTEMS

POLY88 MICROPROCESSOR
A COMPACT, POWERFUL COMPUTER

POLY I/O PROTOBOARD
FULL BUS INTERFACE LOGIC PLUS
ROOM FOR 59-16 PIN WW SOCKETS

VTI/64 VIDEO DISPLAY DRIVER
128x48 GRAPHICS & KEYBOARD INPUT

CROMEMCO

BYTESAVER 8K EPROM USES 2708
D+7AIO D/A A/D 5 MICROSECOND CONVERSION
TV DAZZLER COLOR TV CONTROLLER
JOYSTICKS FOR PROPORTIONAL CONTROL

PLEASE ADD $2.00 POSTAGE & HANDLING
ON ORDERS UNDER $25. SENT UPS
ADD $4.50 POSTAGE & HANDLING ON
ORDERS UNDER $50. SENT POST OFFICE

$10.00 MIMIMUM ORDER
B OF A CARD * COD * PHONE ORDERS

CALIFORNIA RESIDENTS ADD 6% SALES TAX

ADVERTISER INDEX

NOTE: To receive FREE information regarding any product advertised in this issue, simply circle the KEY on the READER SERVICE CARD which matches the advertiser's KEY shown above. Mail the READER SERVICE CARD as soon as possible to receive this service. Do not circle more than 10 KEYS.

We've got it...

You can get it

★ QUALITY Products ★ LOWEST Prices ★ FAST* Service

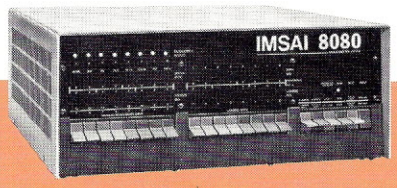
SOUTHWEST TECHNICAL PRODUCTS 6800

Basic Kit with 2 K of Ram

Mfgr.'s List Price	$395.00
OUR PRICE	374.95
add for:	
Shipping & Insurance	00.00
Tax	00.00
Your Cost	**$374.95**

We stock the complete line of SWTP Products. Please call or write for our catalog and price list. You can have the following SWTP options also: • T.V. Typewriter (CT#1024 kit) (259.95), complete with keyboard — cursor — power supply — interface etc. • Cassette Interface (AC-30 kit) (74.95) • Baud conversion kit (13.95).

(TERMS) *All orders will be shipped within 24 hrs. (one business day) from receipt of your order.** (We ship U.P.S. whenever possible). We pay all surface shipping and insurance charges to any point in the lower 48 states. **OUR QUANTITIES ARE LIMITED** — We will send you a notice by mail within 24 hrs. (one business day) of receipt of your order, indicating any items that are out of stock.



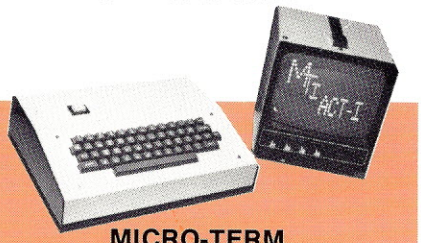
IMSAI 8080

Mfgr.'s List Price	$699.00
OUR PRICE	599.00
add for:	
Shipping & Insurance	00.00
Tax	00.00
Your Cost	**$599.00**

We stock Imsai Ram Boards, interfaces, cables, etc., at similar discounts. Please call or write for our complete catalog and price list. SOME OPTIONS INCLUDE: • A 22 slot motherboard (51.00) • Cooling Fan (27.95 kit) • 1K of Ram (on board expandable to 4K) (57.95 kit) • 1K of Ram chips (for expanding board above) (32.49 kit) • Edge Connectors & Guide (for motherboard & Ram boards, interfaces etc.) (6.75 kit).

MCC

Montana Computer Center
(incorporated)
2512 Grand Avenue
Billings, Montana 59102
(406) 652-2177



MICRO-TERM ACT-1 TERMINAL

Mfgr.'s List Price	$400.00
OUR PRICE	379.95
add for:	
Shipping & Insurance	00.00
Tax	00.00
Your Cost	**$379.95**

Fully Assembled
Features: • 64 characters • 110-9600 Baud • Auto scrolling • RS232c or current loop serial interface • Underline Cursor • 48 hour dynamic burn in. Teletype compatible serial I/O allows direct connection to 6800's, 8080's, & 8800's. 9" Sanyo high resolution monitor also available for $135.00 (shown).

**Certified check or money order assures prompt shipment and delivery. We are required to clear all personal checks. (Personal checks may take up to 3 weeks to clear.)

Stop in to see our new retail store, Billings, Montana at the intersection of I90 & I94 . . . Retail prices are slightly higher in our store.

YOU MUST INCLUDE THIS COUPON TO RECEIVE ABOVE PRICES

Rush me the following items:

- ☐ **IMSAI 8080 Kit** _____
- ☐ **SWTP 6800** _____
- ☐ **MICRO-TERM ACT-1** _____
- ☐ Information on above or below

OPTIONS (Please specify)

_____ _____
_____ _____
_____ _____

MCC Tax & Shipping 00.00

TOTAL ENCLOSED _____

Name _____

Address _____

City _____ State _____ Zip _____

Bank Cards (BAC & M.C. accepted)

☐ BankAmericard ☐ Master Charge

Acct. No. _____ Exp. Date _____

Signature _____

Montana Computer Center, 2512 Grand Avenue, Billings, Montana 59102

SWTPC 6800

- ★ COMPLETE WITH 2K OF MEMORY
- ★ SERIAL INTERFACE
- ★ STANDARD ROM MONITOR (Motorola MC 6830L7)
- ★ 4K AND 8K BASIC AVAILABLE

Always the best value in hardware and now an outstanding selection of software too. What more could you want in a computer system? For less than four hundred dollars you get everything you need—ready to connect to a terminal and go to work. No surprises no funny business, just good reliable hardware in a very practical system that may be expanded to meet almost any later need.

Memory and interfaces are not extra cost items in our system. A standard Motorola MIKBUG® ROM monitor makes the system completely compatible with Motorola software and eliminates any need for console switches and light. Data may be entered from the terminal in convenient hexidecimal form. The power supply is adequate to operate a fully expanded system with up to 24K of memory and up to eight (8) interfaces—simultaneously.

See the 6800 and our peripheral equipment at your nearest dealer, or write for a complete description.

MP-68 COMPUTER KIT—with serial interface, 2k of memory and ROM monitor **$395.00 ppd**

® Motorola

Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, Texas 78216

CIRCLE 2